

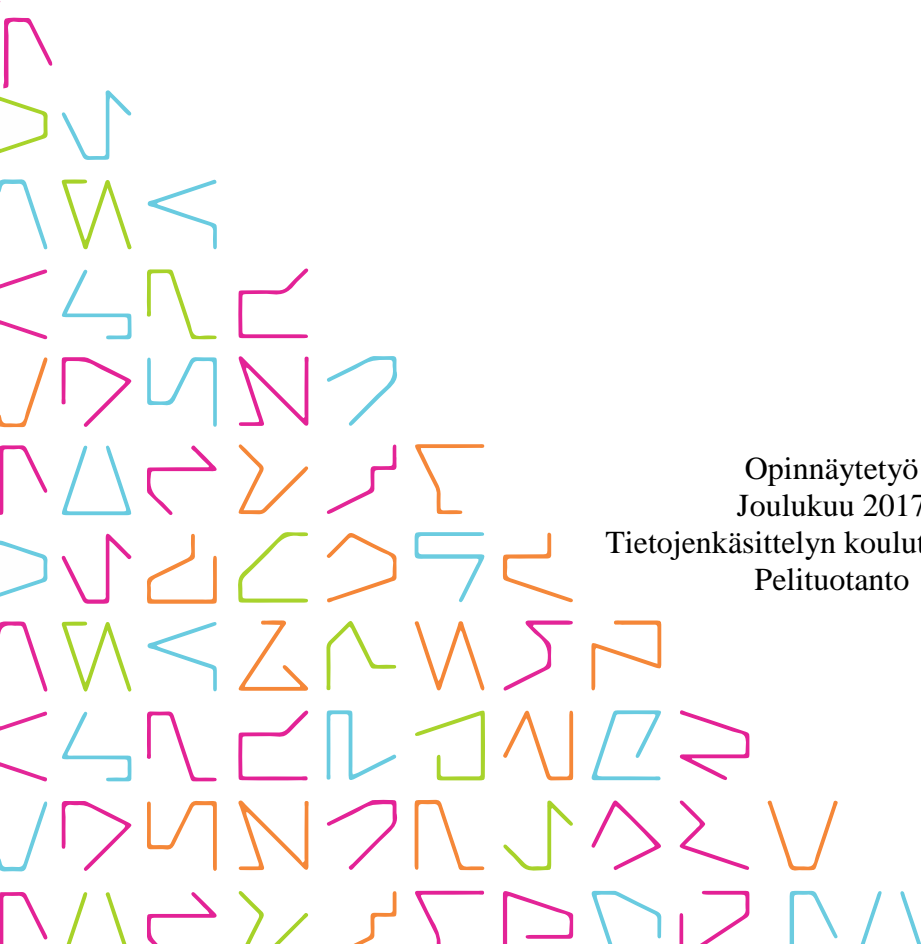


TAMPEREEN
AMMATTIKORKEAKOULU

TEKSTUROIINTI UNITY-PELIMOOTTORILLA TOTEUTETUSSA KOLMANNEN PERSOONAN PC-PELISSÄ

Janne Pauni

Opinnäytetyö
Joulukuu 2017
Tietojenkäsittelyn koulutusohjelma
Pelituotanto



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely
Pelituotanto

PAUNI, JANNE:

Teksturointi Unity-pelimootorilla toteutetussa kolmannen persoonan PC-pelissä

Opinnäytetyö 102 sivua
Joulukuu 2017

Tämän opinnäytetyön tavoitteena oli tutkia teksturointia Fragment Production -yrityksen Unity-pelimootorilla toteutetussa PC-pelissä. Opinnäytetyön projektiosuuden tarkoituksena oli tuottaa ohjelinjojen mukaista, laatukriteerit täyttävää 3D- sekä tekstuurigrafiikkaa Fragment Productionin peliprojektiin Rescue 2: Everyday Heroes ja arvottaa projektin onnistumista etenkin teksturoinnin osalta.

Tekstuureilla eli tekstuurikartoilla tarkoitetaan tietokonegrafiikassa 3D-mallien pinnoilla käytettäviä kuvatiedostoja, jotka antavat kappaleille pintarakenteen. Teksturointiprosessissa 3D-malli pinnoitetaan bittikarttakuvalla, johon on piirretty pintarakenteen ominaisuudet.

Tekstuurikarttoja on olemassa erityyppisiä, joilla kaikilla on oma tarkoituksensa materiaalien fysikaalisten ominaisuuksien välittämisessä katsojalle. Tekstuureita on mahdollista luoda erilaisilla ohjelmistoilla ja tekniikoilla, joiden valinta on yleensä projektikohtaista. Tekstuurikartoilla voidaan luoda erilaisia visuaalisia tyylejä peleille. Tekstuurien tyyli riippuu pelin lajityypistä ja projektin tyylivalinnoista.

Rescue 2: Everyday Heroes on palomiestoimintaan keskittyvä simulaatiopeli. Tekstuurigrafiikassa pelin tunnelmaa pyrittiin vahvistamaan realismia ja tyyliä ulkoasua sekoittavalla teksturointityylillä. Pelin tekstuurit toteutettiin käyttämällä Photoshop-kuvankäsittelyohjelmaa, sen NVIDIA Texture Tools for Adobe Photoshop -liitännäistä sekä CrazyBump ja Quixel Suite 2 -ohjelmistoja.

Projektin graafinen puoli onnistui melko hyvin, joskin uudemman teknologian hyödyntäminen olisi parantanut lopputulosta huomattavasti. Kuten yleisesti muissakin vastaavanlaisissa projekteissa, grafiikkapuolen ongelmakohdiksi muodostuivat artistitiimin pieni koko työmäärään nähden sekä projektin tiukka aikataulu. Parannusehdotuksena tuleviin projekteihin huomattiin grafiikan tuottamisen ulkoistaminen alihankkijalle, mikä vähentäisi artistien työkuormaa.

Asiasanat: tekstuuri, tekstuurikartta, teksturointi, tietokonegrafiikka, pelinkehitys

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems
Game Production

PAUNI, JANNE:
Texture Mapping in a Third Person PC Game Made with Unity

Bachelor's thesis 102 pages
December 2017

The aim of this thesis was to study the texturing in a PC game made with the Unity game engine by Fragment Production. Moreover, there were two practical goals: to produce 3D and texture graphics for Fragment Production's game project Rescue 2: Everyday Heroes that would meet the quality criteria, and to evaluate the success of the project especially regarding the texturing.

Textures or texture maps are image files used in computer graphics on the surfaces of 3D models, which give the model surface a texture. In the texturing process, the 3D model is coated with a bitmap image that contains the features of the object's surface structure.

There are different types of texture maps, each of which has its own purpose to convey the physical properties to the viewer. Texture maps can be created with different software and techniques, the choice of which is usually specific to the project. With texture maps it is possible to create different visual styles for games. The style of the textures depends on the genre of the game and which style has been selected for the project.

Rescue 2: Everyday Heroes is a simulation game focusing on firefighting. The aim with texture graphics was to enhance the atmosphere of the game by using a style that mixed realism and stylized texture art. Texture maps in the project were created using the Photoshop image editing software, the NVIDIA Texture Tools for Adobe Photoshop plugin, CrazyBump, and Quixel Suite 2.

The goals set for the game graphics were reached quite successfully, though using newer technology would have produced a more improved outcome. As seen in other such projects of the same scale, the workload and schedule were quite disproportionate to the small size of the art team, which caused some difficulties in reaching the optimal results. A solution for future projects would be to outsource the graphics production to a subcontractor which would reduce the workload of artists in the project.

Key words: texture, texture map, texture mapping, computer graphics, game development

SISÄLLYS

1	JOHDANTO.....	7
2	3D-MALLINNUKSEN PERUSTEITA	8
2.1	3D-mallit.....	8
2.2	Varjostusmallit.....	9
2.3	UV-kartoitus	10
2.4	Materiaalit ja varjostimet	13
2.5	Suunnitelman tekeminen ja hyvät käytännöt	15
3	PBR -STANDARDI.....	18
4	MITÄ OVAT TEKSTUURIKARTAT?	21
5	TEKSTUURIKARTTATYYPIT	23
5.1	Diffuusiokartta	23
5.2	Perusvärikartta	24
5.3	Kohoumakartta.....	26
5.4	Normaalikartta	27
5.5	Siirtymäkartta.....	29
5.6	Spekulaarisuuskartta	30
5.7	Kiilto-/karheuskartta	32
5.8	Metallikartta.....	33
5.9	Ympäristön okkluusiokartta.....	35
5.10	Emissiivisyyskartta	37
5.11	Läpikuultavuuskartta	38
5.12	Peittomaskit	39
5.13	Tekstuuriatlakset	40
5.14	Tiheyskartta	42
5.15	MIP-kartoitus	43
5.16	Ympäristökartta	45
6	TEKSTUURIEN VISUAALISET TYYLIT	47
6.1	Realistinen teksturointi	47
6.2	Käsinmaalattu teksturointi	49
6.3	Puolirealistinen teksturointi	51
6.4	Abstraktit teksturointityylit.....	53
6.5	Proseduraaliset tekstuurit.....	53
6.6	Tekstuurien tyyli Rescue 2: Everyday Heroes -pelissä	54
7	KAMERAKUVAKULMAN VALINTA.....	57
7.1	Ensimmäinen persoona	58
7.2	Kolmas persoona.....	59

7.3	Muut kuvakulmat	60
7.4	Kamerakuvakulma Rescue 2: Everyday Heroes -pelissä.....	62
8	TEKSTUURIKARTTOJEN LUOMINEN ERI OHJELMILLA	65
8.1	Adobe Photoshop	65
8.2	NVIDIA Texture tools for Adobe Photoshop	67
8.3	CrazyBump	68
8.4	Quixel Suite 2	69
8.5	Knald.....	71
8.6	XNormal	72
8.7	Substance Designer, Substance Painter ja Substance B2M	72
8.8	BodyPaint 3D.....	74
8.9	Mari.....	74
9	TEKSTUURIKARTAN LUOMINEN RESCUE 2 -PELISSÄ	76
9.1	Dokumentin luominen ja alkuvalmistelut.....	77
9.2	Pohjavärialueiden luominen.....	78
9.3	Varjojen lisääminen	79
9.4	Väriliukujen lisääminen	80
9.5	Saumojen maalaaminen	81
9.6	Korostuksien maalaaminen.....	82
9.7	Yksityiskohdat ja sävymuutokset	83
9.8	Lian lisääminen.....	84
9.9	Tekstien lisääminen	85
9.10	Lopputuloksen tarkastelu ja tekstuurin tallentaminen	85
9.11	Spekulaarisuuskartan luominen	86
9.12	Normaalikartan luominen	87
9.13	Pelimoottoriin vieminen	88
	POHDINTA	91
	LÄHTEET.....	94

ERITYISSANASTO

Assetti Pelin luomisessa käytettävä grafiikkaobjekti, kuten tekstuurikartta tai 3D-malli.

Baking Menetelmä, jossa kolmiulotteisen mallin geometriaa käytetään 3D-ohjelmassa piirtämään bittikarttakuva mallin erilaisista ominaisuuksista, kuten pinnanmuodoista. Suomeksi menetelmää kutsutaan leipomiseksi.

High poly Korkeasta tahkomäärästä koostuva 3D-malli, jota käytetään yksityiskoh-
tien luomiseen matalaresoluutioisten 3D-mallien pinnoille.

Low poly Matalasta tahkomäärästä koostuva 3D-malli, jota käytetään reaaliaikaises-
sa kuvantamisessa.

Pelimoottori Pelikehityksessä käytettävä ohjelma, joka sisältää ominaisuuksia, kuten
sisäänrakennetun fysiikka- ja valaistusmoottorin.

Polygoni Kolmen tai useamman kärjen muodostama kaksiulotteinen muoto. Yhdes-
sä useampien polygonien kanssa ne muodostavat 3D-mallien pinnanmuodot.

Renderöinti Tietokoneen suorittama laskutoimitus, jonka avulla se piirtää 3D-malleista
väritetyn ja valaistun kuvan tietokoneen ruudulle. Termistä käytetään myös nimityksiä
kuvantaminen ja hahmontaminen.

Verteksi Kärki tai avaruuspiste, joka yhdessä särmien ja muiden verteksien kanssa
muodostaa 3D-mallin tahkot.

1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on esitellä peligrafiikassa käytettyjä tekniikoita sekä tekstuurikarttatyyppejä. Opinnäytetyön projektiosuudessa esitellään, miten tekstuurikarttojen luominen projektissa tapahtui ja mitä työkaluja niiden tekemisessä käytettiin. Projektiosuuden lisäksi pyrittiin arvioimaan pelin visuaalisen ilmeen onnistumista sekä löytämään kehitysehdotuksia seuraavien projektien graafisen puolen parantamiseksi.

Työtehtäväni projektissa keskittyivät enimmäkseen ympäristöön liittyvien assettien teksturointiin ja mallintamiseen, joten teksturointiprosessia ja tekstuureihin liittyviä ominaisuuksia pyritään tarkastelemaan ympäristön teksturoinnin näkökulmasta. Opinnäytetyö on toteutettu yhteistyössä Fragment Productionin kanssa.

Fragment Production on Tampereella sijaitseva pelinkehitysstudio, joka on perustettu vuoden 2012 helmikuussa. Studio koostuu lähes kolmestakymmenestä alan ammattilaisesta. Fragment Production on erikoistunut tekemään strategia- sekä simulaatiopelejä. Yrityksen uusi julkaisu on vuoden 2017 tammikuussa julkaistu Urban Empire PC-peli, joka on uniikki yhdistelmä kaupunginrakennusta ja poliittista simulaatiota.

Vuonna 2013 studio julkaisi ensimmäisen PC-pelinsä "Rescue: Everyday Heroes". Rescue-pelisarja jatkui mobiiliversiolla pelistä vuonna 2015, nimellä "Rescue: Heroes in Action" sekä jatko-osalla ensimmäiselle PC-pelille nimellä "Rescue 2: Everyday Heroes".

Rescue 2 on sammutus- ja pelastustoimintaan keskittyvä yksinpeli, joka yhdistelee simulaatiota, managerointia ja reaaliaikastrategiaa. Pelissä pelaaja omaksuu palopäällikön roolin, jonka tehtävänä on johtaa paloasemia eri puolilla kaupunkia. Pelaajan täytyy hankkia kalustoa sekä työntekijöitä paloasemille, manageroida näitä ja komentaa palomiehiä sekä ensihoitajia vaihtelevissa sammutus- ja pelastustehtävissä ympäri kuvitteellista keskieurooppalaista Belvitsian kaupunkia. Reaaliaikastrategiapeleille tyypilliseen tapaan pelaaja näkee tapahtumat viistossa kulmassa maaton yläpuolelta katsottuna. Peli on toteutettu käyttäen Unity-pelimootoria.

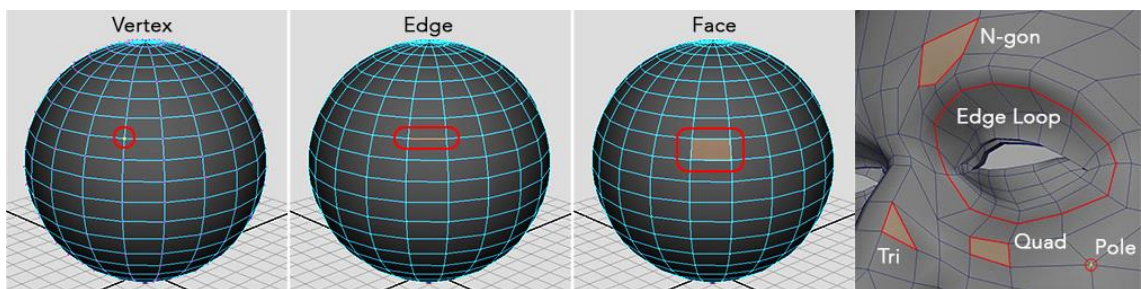
2 3D-MALLINNUKSEN PERUSTEITA

Kolmiulotteisesti piirrettävä grafiikka on yleistynyt nykypäivän standardiksi tietokonepeleissä ja nykyään moni julkaistava peli käyttää kolmiulotteista grafiikkaa. On hyvä ymmärtää kolmiulotteisen mallinnuksen perusteita, jotta tekstuurien käyttötarkoituksesta ja toiminnasta saa paremman käsityksen. Tämä auttaa artistia teksturointiprosessissa, kun hän tietää, miten tekstuurit toimivat 3D-mallien kanssa.

2.1 3D-mallit

3D-malli on matemaattinen esitystapa mistä tahansa kolmiulotteisesta kappaleesta esitettyä kolmiulotteista ympäristöä tukevassa ohjelmassa. Toisin kuin kaksiulotteiset kuvat, 3D-malleja on mahdollista tarkastella mistä tahansa kuvakulmasta niihin erikoistuneissa ohjelmistoissa. Tämän lisäksi 3D-mallien kokoa voidaan muuttaa, niitä voidaan pyörittää ja niitä voi muokata vapaasti. 3D-mallin luomista tai sen muokkaamista kutsutaan 3D-mallinnukseksi. (Slick 2016.)

Peligrafiikasta puhuttaessa 3D-mallit koostuvat yleisimmin kärjistä (vertices), särmistä (edges) ja tahkoista (faces) (Slick 2016) Kuvassa 1 on esiteltyä 3D-mallien osat sekä eri mallisista tahkoista käytetyt termit.



KUVA 1. 3D-mallin osat ja eri mallisten tahkojen nimitykset (Smith 2013)

Kärjet ovat perustavin osa 3D-mallista. Ne ovat yksittäisiä pisteitä kolmiulotteisessa avaruudessa ja jokaisella niistä on oma koordinaattinsa. 3D-mallia renderöitäessä särmät ovat yleensä näkymättömiä. Särmät yhdistävät kärkiä luomalla niiden välille suoran janan ja niitä käytetään tahkojen luomisessa. Kärkien tapaan ne ovat yleensä näkymät-

tömiä, kun 3D-mallia renderöidään. Tahkot ovat kolmen, neljän tai useamman kärjen muodostamia pintoja, joita särmät ympäröivät. Tahkot muodostavat 3D-mallin varsinaisen pinnan ja ovat näkyvä osa 3D-mallia renderöitäessä. Kolmen kärjen muodostamaa tahkoa kutsutaan kolmioksi (triangle, lyh. tri) ja neljän kärjen muodostamaa tahkoa nelikulmioksi (quadrable, lyh. quad). Useamman, kuin neljän kärjen muodostamia tahkoja kutsutaan n-goneiksi (n-gon). (Blender Manual 2015.) N-goneille ei ole vakiintunutta suomenkielistä termiä.

Hyvässä mallinnustavassa 3D-mallin tahkot koostuvat kolmesta tai enintään neljästä särmästä. Tahkojen määrästä käytetään nimitystä poly-count, joka tarkoittaa suomeksi polygonien yhteenlaskettua määrää. Hyvät mallintajat pyrkivät tehokkuuteen ja hyvään järjestykseen tahkoja luodessaan, yrittäen pitää niiden määrän mahdollisimman alhaisena. Tahkojen tiheydestä käytetään nimitystä resoluutio. Isomman resoluution 3D-mallit (high-poly) näyttävät piirrettäessä yleensä tasaisemmilta, kun alhaisen resoluution 3D-mallit (low-poly). (Slick 2016.)

Tahkomallinnuksen ohella toinen tapa muodostaa 3D-malleja on NURBS-mallinnus. NURBS on lyhenne sanoista non-uniform rational Bezier-spline, joka tarkoittaa suomeksi epäyhdenmukaista rationaalista B-käyräviivainta. NURBS-mallinnuksessa 3D-mallin pinta luodaan käyräviivaimia käyttämällä. Pinnan luomiseksi artisti piirtää kolmiulotteiseen tilaan kaksi tai useampia käyriä, joiden muotoa hallitaan liikuttamalla hallintakahvoja x-, y-, ja z-akseleilla. Tietokoneohjelma laskee käyräviivainten välissä olevan tilan ja luo niiden väliin sileän pinnan. NURBS-pinnoilla on korkein matemaattinen tarkkuus ja tämän takia niitä käytetään yleensä tekniikan alan mallinnuksessa, kuten automuotoilussa. (Slick 2016.)

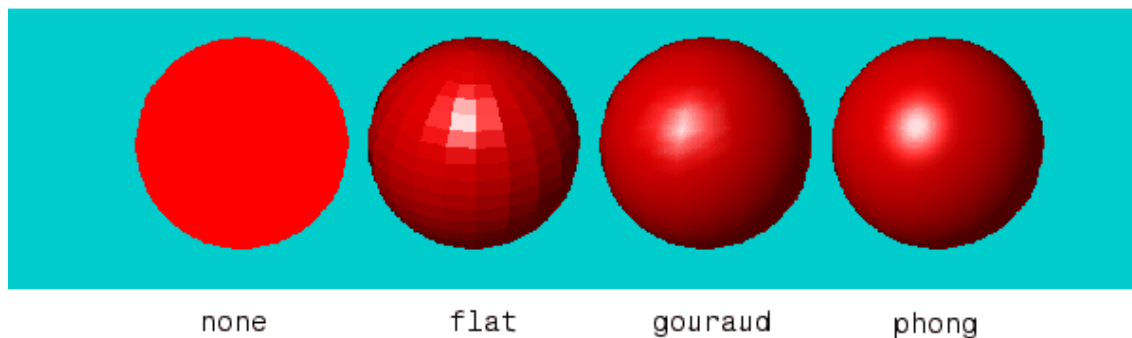
2.2 Varjostusmallit

Varjostusmallit ovat tekniikoita, joiden avulla 3D-mallien pinnalle luodaan paikallinen valaistusmalli (Owen 1999). Käytännössä tämä tarkoittaa sitä, että varjostinohjelma muuttaa 3D-mallin pintoja esittävien pikseleiden tummuutta syvyysvaikutelman aikaansaamiseksi. Tosiaikaisessa kuvantamisessa on nykyisellään käytössä kolme erilaista varjostusmallia (Kahrama 2001).

Flat-varjostus on yksinkertaisin käytössä olevista varjostusmalleista. Siinä jokaiselle 3D-mallin tahkolle lasketaan vain yksi väri. Flat-varjostus ei ole kovin realistinen, mutta tietokoneen laskentatehon kannalta se on nopein tapa kuvantaa syvyyttä. (Owen 1999.)

Gouraud-varjostuksessa jokaiselle 3D-mallin kärjelle lasketaan valaistuksen määrän intensiteetti, jonka perusteella kolmioiden pinnat sävytetään lineaarisesti interpoloimalla. Tällä tavalla voidaan välttää terävät muutokset tahkojen reunoilla. (Owen 1999.) Gouraud-varjostuksen avulla on mahdollista luoda sulavampia siirtymiä pyöreiden objektien väreissä. Huonona puolena tässä mallissa on kuitenkin sen huono kyky varjostaa materiaaleja, joissa on peiliheijastuksia. (Jaggo, Luik & Tunnel 2017.)

Phong-varjostus on samankaltainen, kuin Gouraud-varjostus, mutta eroaa siitä siten, että kärkien sijaan tahkot sävytetään niiden normaalien perusteella. Tämän takia peiliheijastukset voidaan laskea paljon tarkemmin, kuin Gouraud-varjostuksessa. Gouraud-varjostukseen verrattuna Phong-malli on huomattavasti raskaampi laskentatehon kannalta. (Owen 1999.) Kuvassa 2 on esitettynä eri varjostusmallit pallon muotoisen objektin pinnalla sekä pallo ilman valaistusta (None, Flat, Gouraud, Phong).



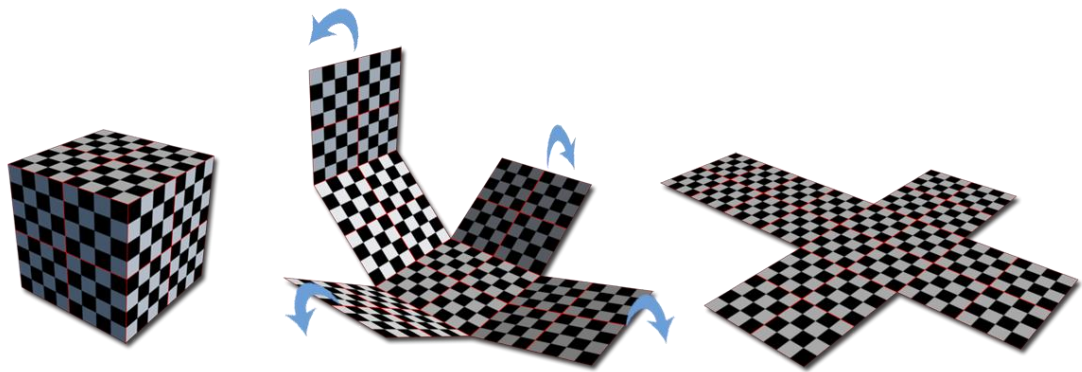
KUVA 2. Varjostinmallit (Department of Radio Engineering 2017)

2.3 UV-kartoitus

Ennen kuin 3D-mallia on mahdollista teksturoida, on sen geometria levitettävä tasoon. Kaikista joustavin tapa kartoittaa kaksiulotteinen kuva kolmiulotteisen objektin päälle on tekstuuri- eli UV-kartoitus (engl. UV-mapping). Tässä prosessissa kolmiulotteisen avaruuden X-, Y- ja Z-koordinaatit puretaan tasoon kaksiulotteiselle pinnalle. Kuvatie-

dostossa olevat pikselit kartoitetaan näin 3D-mallin pinnalle, jolloin ne näkyvät erilaisina yksityiskohtina objektin tahkoilla. UV-kartoituksen avulla on mahdollista luoda realistisempia ja tarkempia pintarakenteita 3D-mallien pinnoille, kuin muilla tekniikoilla. (Blender 2.79 Manual 2017.)

Paras havainnollistus UV-kartoituksen ymmärtämiseen on pahvilaatikon leikkaaminen osiin (kuva 3). Pahvilaatikko on kolmiulotteinen objekti, kuten 3D-ohjelmassa nähtävä kuutio. Jos laatikon leikkaa auki sen saumoista, sen pystyy levittämään tasoon. Kun tasaista laatikkoa tarkastellaan tasossa, voidaan vaakatasossa olevaa akselia kutsua U-akseliksi ja pystysuuntaista akselia V-akseliksi. UV-kartoituksessa U- ja V-kirjaimia käytetään viittaamaan näihin tekstuuriavaruuden koordinaatteihin, X- ja Y-koordinaattien sijaan, joita käytetään Z-akselin ohella kolmiulotteisesta tilasta puhuttaessa. Kun laatikko kootaan takaisin alkuperäiseen muotoon, tasossa olleet UV-koordinaatit siirretään laatikon pinnalle kolmiulotteiseen avaruuteen. (Blender 2.79 Manual 2017.) Kuvassa 3 on esitettyä kuution 3D-malli, joka avataan tasoon.



KUVA 3. 3D-mallin avaaminen tasoon. (Pytte 2016)

3D-mallinnusohjelmista löytyy UV-kartoitusta varten soveltuva editori, josta löytyy monia työkaluja, millä kolmiulotteinen pinta kartoitetaan kaksiulotteiseen muotoon. Yksinkertaisimmat niistä ovat erilaisia projisointitapoja kuten taso- ja lieriöprojisointi, joita käytetään tavanomaisten muotojen kartoittamiseen. Monimutkaisempien muotojen kanssa voidaan käyttää kehittyneempiä työkaluja, joilla on tarkempia käyttötarkoituksia. (Blender 2.79 Manual 2017.) Projisointeja ja muita työkaluja käyttämällä 3D-mallin tahkot kartoitetaan UV-alueiksi ja asetellaan yksilöllisille paikoilleen editorialueen sisälle (Slick 2016). UV-kartalla tarkoitetaan projisoiduista ja asetelluista UV-alueista

tehtyä pohjapiirrosta (Brinck 2009). Tekstuuriartisti käyttää UV-karttaa myöhemmin mallina kuvankäsittelyohjelmassa tekstuurikarttaa piirtäessään (Slick 2016).

Teksturointiprosessin helpottamiseksi on tärkeää, että UV-alueiden asettelu UV-kartan sisällä on tehty hyvin. UV-kartoitukselle ei ole olemassa tiukkoja sääntöjä, mutta työkentelyn helpottamiseksi on olemassa muutamia ohjenuoria, joita artistin on syytä noudattaa. UV-kartan on hyvä koostua mahdollisimman vähistä UV-alueista. Tämä tekee tekstuurin piirtämisestä helpompaa ja auttaa hahmottamaan, mitkä UV-alueet vastaavat mitään tahkoa 3D-mallissa. Bittikartan venymistä 3D-mallin tahkoilla on myös syytä välttää, joten sen kunnolliseen projisointiin on kiinnitettävä huomiota. Venymistä tapahtuu silloin, kun UV-alue ei vastaa kooltaan tai muodoltaan sitä vastaavaa 3D-mallin tahkoa. Venymisen lisäksi päällekkäin olevia UV-alueita on vältettävä, koska bittikartassa samalle alueelle ei voi piirtää kahta erilaista yksityiskohtaa. Poikkeuksena tähän ovat päällekkäin asetellut UV-alueet, joiden käyttö on yleistä silloin, kun 3D-malli on symmetrinen tai siinä on useita toistuvia yksityiskohtia.

UV-kartan on hyvä käyttää tekstuurialue mahdollisimman tehokkaasti hyödyksi, koska kohdat, joilla ei ole UV-alueita ovat hukkatilaa. Huolellisella asettelulla hukkatilaa on mahdollista vähentää, jolloin UV-alueille saadaan parempi resoluutio. UV-alueiden tulisi olla myös suhteessa saman kokoisia, kuin niitä vastaavien tahkojen. Muutoin pienempää resoluutiota käyttävät alueet näyttävät rakeisemmilta isompiresoluutioisten kohtien rinnalla. Poikkeuksena tähän on esimerkiksi pelihahmon kasvot, joille halutaan tarkemmat yksityiskohdat, kuin muulle vartalolle. Näiden ohjenuorien avulla artisti luo mahdollisimman optimaalisen ja käytännöllisen ratkaisun teksturoinnin ja tekstuurien toimivuuden kannalta. UV-kartoitukseen ei ole olemassa yhtä oikeaa tapaa, joten artistin on valittava tilanteeseen sopivin tapa aikamäärän sallimissa rajoissa. (Brinck 2009.)

UV-kartoituksen avulla artistin on mahdollista luoda erityyppisiä sommitteluita tekstuurille erilaisia käyttötarkoituksia varten. Laatoittuvalla tekstuurilla (engl. tiling texture) tarkoitetaan saumatonta tekstuuria, jota on mahdollista toistaa pinnoilla. Laatoittuvan tekstuurin on oltava saumaton joko toisella tai molemmilla U- ja V-akseleista, jotta se toimii visuaalisesti. Laatoittuvia tekstuureita käytetään yleisesti peleissä kattamaan laajoja pintoja, jotka eivät tarvitse erityisiä yksityiskohtia, kuten maa-alueita, tiiliseiniä tai maisemaa. Uniikit tekstuurit (engl. unique texture) puolestaan eivät sisällä toistuvia elementtejä vaan sen sijaan useita omille paikoilleen sommiteltuja uniikkeja UV-alueita.

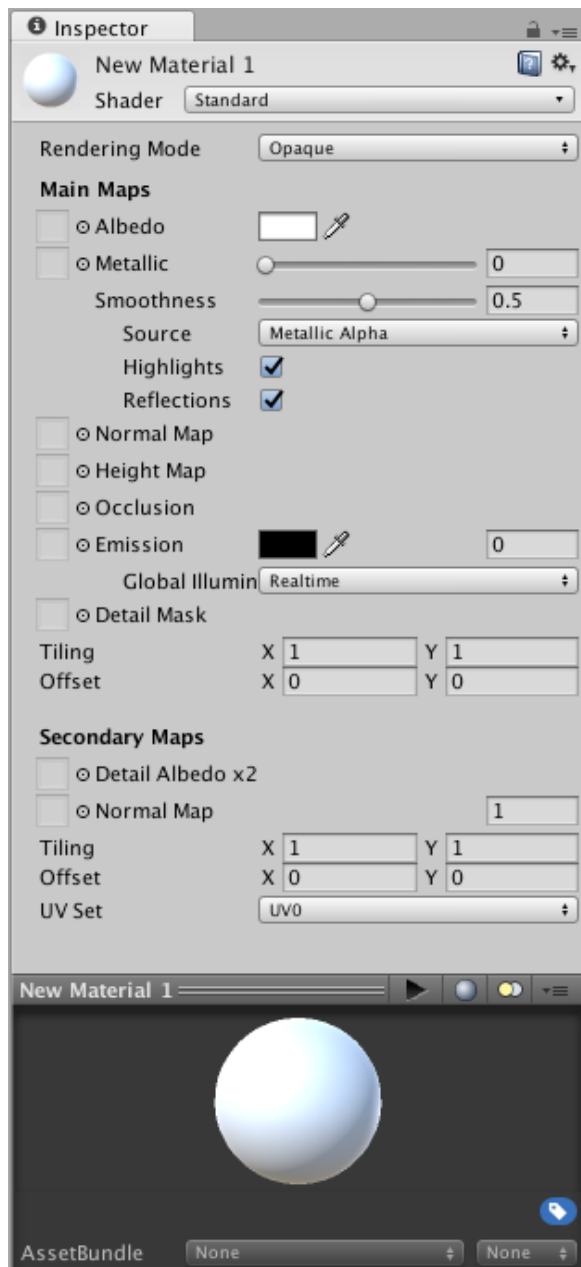
Tällaista sommittelua käytetään yleisesti pelihahmoissa ja objekteissa, jotka tarvitsevat erityisiä yksityiskohtia. Uniikkeja ja laatoittuvia elementtejä käyttävä tekstuuri sen sijaan yhdistelee yhdellä tai kahdella akselilla laatoittuvia elementtejä sekä omille alueille varattuja UV-alueita samalla tekstuurikartalla. (Gahan 2012, 7-8.)

2.4 Materiaalit ja varjostimet

Ilman varjostimia 3D-mallit eivät näyttäisi erityisen näyttäviltä. 3D-mallien muodon sijasta materiaalit ja varjostimet vaikuttavat niiden ulkonäköön. (Slick 2016.)

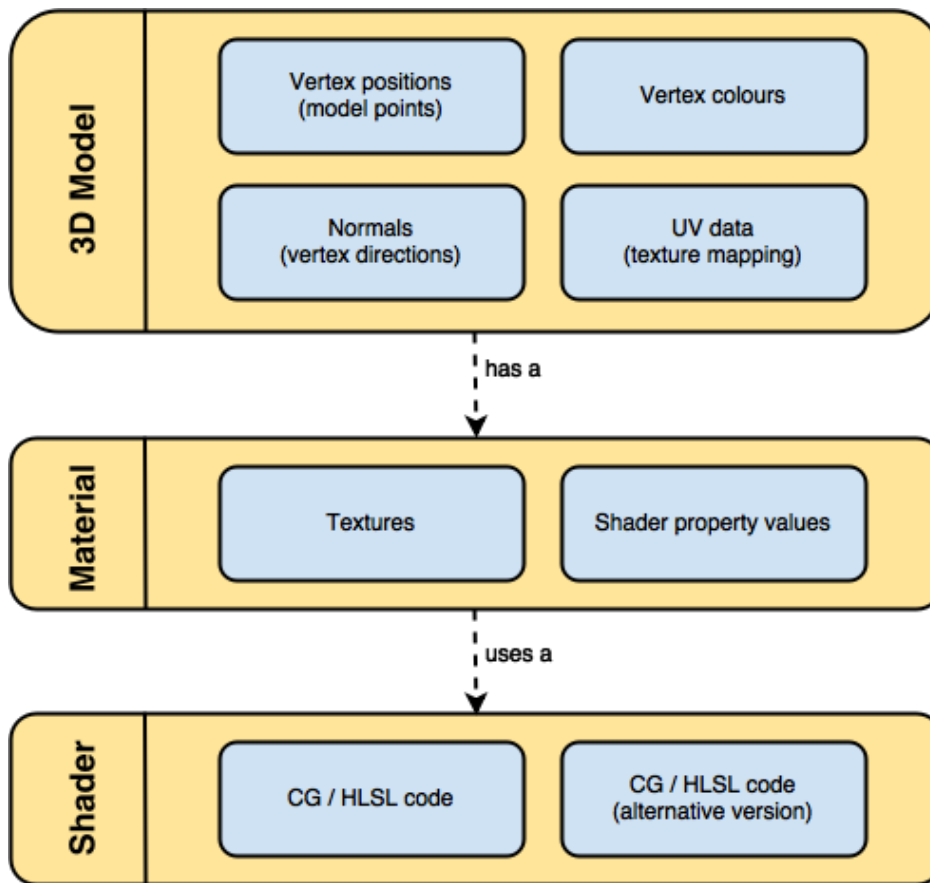
Yksinkertaisesti selitettynä varjostin (engl. shader) on pienikokoinen ohjelma, joka prosessoi graafisia efektejä reaaliajassa (Gahan 2012, 94-95). Ne sisältävät matemaattiset laskut ja algoritmit jokaisen 3D-mallin pikselin värin piirtämiseksi (Unity Manual 2017). Varjostimien avulla esimerkiksi heijastukset voivat käyttäytyä tosiaikaisesti kuvan liikuessa sen sijaan, että ne piirrettäisiin staattisena 3D-mallin pintaan. Varjostimien kyky käsitellä yksittäisiä pikseleitä tekee niistä erityisen tehokkaita, mutta samalla vaativat tietokoneelta paljon laskentatehoa. Varjostimia voidaan käyttää useiden monitukaisten materiaalien ja erikoistehosteiden, kuten tulen, varjojen, veden ja heijastusten kuvantamiseen. Varjostimet ovat niin joustavia, että varjostinohjelmoijan on mahdollista kirjoittaa niille lähes loputtomasti erilaisia visuaalisia efektejä. Ohjelmoijat ovat usein koodanneet varjostimet valmiiksi peliprojektiin, jolloin artistin tehtäväksi jää säätää varjostimen asetuksia halutun lopputuloksen aikaansaamiseksi. Artistin rooli varjostimien kanssa työskennellessä koostuu yleensä niiden käyttämien kuvatiedostojen luomisesta, varjostimen asetusten muokkaamisesta ja lopputuloksen ulkonäön arvioinnista. Vaikka artisti ei olekaan tekemisissä varjostimien koodin kanssa, on tämän ymmärrettävä, miten ne toimivat. (Gahan 2012, 95, 98-99.)

Varjostimet ja materiaalit ovat tiiviisti sidoksissa toisiinsa. Varjostimien toimintaa hallitaan materiaalien kautta, jotka taas ovat linkitettyinä kuvannettaviin 3D-malleihin. Materiaalin avulla valitaan käytettävä varjostin, sekä muutetaan sen asetuksia. Kuva 4 esittää Unity-pelimoottorin käyttämää materiaalia.



KUVA 4. Unity-pelimoottorin materiaali, joka käyttää Standard-varjostinta (Unity Technologies 2017)

Materiaalien avulla voidaan määrittää, mitä väriä ja bittikarttakuvia varjostimet käyttävät, sekä mitä muita resursseja 3D-mallin kuvantamiseen tarvitaan. (Unity Manual 2017.) Varjostimen koodi määrittelee yhden tai useamman tekstuurityypin, mitä materiaali voi käyttää ja materiaalin käyttöliittymän avulla artistin on mahdollista lisätä omat tekstuurinsa varjostimen käytettäväksi. Materiaali voi käyttää yhtä varjostinta ja valittu varjostin määrittää sen, miten ja kuinka paljon materiaalia on mahdollista muokata. (Unity Manual 2017.) Kuva 5 havainnollistaa 3D-mallien, materiaalien ja varjostimien toimintaa Unity-pelimoottorissa.



KUVA 5. 3D-mallien, materiaalien ja varjostimien toiminta Unityssä (Unity Technologies 2017)

2.5 Suunnitelman tekeminen ja hyvät käytännöt

Peliobjektien tekemistä aloitettaessa artistin on suositeltavaa omaksua tiettyjä tapoja, jotka helpottavat työskentelyä ja parantavat työn lopputulosta. Suunnitelman tekeminen ja asioiden pitäminen järjestyksessä ovat molemmat osa hyvää työnkulkua. Osa näistä tavoista on kriittistä hallita itsenäisessä työssä sekä työskenneltäessä muiden artistien kanssa projekteissa. (Chopine 2012, 16.)

Projektia aloitettaessa on tärkeää päättää, mitä siihen ollaan tekemässä ja miltä lopputuloksen halutaan näyttävän. Esikuvien valitseminen ja yksityiskohtien tarkkuuden päättäminen ohjaavat projektin graafista puolta. Näiden sivuuttaminen voi johtaa epäyhteoiseen lopputulokseen ja tuhlaa paljon aikaa. Käytettävissä oleva aika pitää olla selvillä, sillä se auttaa suunnittelemaan, kuinka paljon sisältöä projektiin on mahdollista tuottaa. (Gahan 2012, 15.)

Yksi hyvä tapa selvittää artistitiimille, millaista tyyliä projektiin ollaan hakemassa, on koota niin sanottu mood board. Mood board -termillä tarkoitetaan taulua, johon kootaan referenssikuvia siitä, miltä lopullisen graafisen ilmeen halutaan näyttävän. Referenssikuvat voivat olla luonnoksia, Internetistä löydettyjä kuvia, kuvankaappauksia elokuvista tai mitä tahansa, minkä ulkonäkö, sävy tai muoto kuvaavat suunniteltua lopputulosta. (Gahan 2012, 15.)

Ennen työhön ryhtymistä pitää selvittää, kuinka tarkkaa 3D-mallien geometriasta halutaan, tarvitaanko 3D-mallista erilaisia versioita, mitä tekstuurikarttatyppejä tarvitaan ja kuinka isoja tekstuurikarttojen koosta halutaan tehdä. Projektissa työskenneltäessä on tärkeää, että jokainen artisti tuottaa yhteneväisiä asetteja projektiin, niin mittakaavan kuin tyylin osalta. (Gahan 2012, 16.) Asettien ominaisuuksien lisäksi on selvitettävä, millaisia tekniikoita projektissa halutaan käyttää ja onko uusien työtapojen opettelulle tai ohjelmien hankinnalle tarvetta. Tällaisten asioiden selvittäminen etukäteen nopeuttaa työskentelyä ja vähentää kesken projektia ilmeneviä ongelmia, kun niitä voidaan ennakoida jo ennen työhön ryhtymistä.

Kun projektiin tuotettava sisältö ja sen laatuvaatimukset ovat tiedossa, on tehtävä suunnitelma niiden tekemiseksi. Suunnitelman avulla voidaan välttää liiallinen työmäärä ja varmistaa, että asetettuihin aikarajoihin voidaan päästä ajallaan. Sisällön purkaminen yksittäisiin työtehtäviin ja niiden tekemiseen kuluvan ajan arvioiminen auttaa projektin mittakaavan hahmottamisessa. Jos työmäärä on arvioitaessa liian suuri, suunnitelmia voidaan tehdä uusiksi ja projektin laajuutta voidaan muuttaa helposti ennen kuin työtä aloitetaan. (Gahan 2012, 17-18.)

Ennen asettien tuottamisen aloittamista artisti tarvitsee konseptikuvia mallinnettavista peliobjektista. Konseptikuvat voivat olla kuvaesimerkkejä, luonnoksia tai ortografisia kuvituksia. Ilman selkeitä konseptikuvia mallintamista ei kannata aloittaa, sillä muutosten tekeminen suunniteluvaiheessa on helpompaa, kuin puolivälissä työtä. (Gahan 2012, 17.)

Pelimaailmaa rakennettaessa artistin luomien asettien määrä tulee kasvamaan huomattavasti projektin edetessä. Käyttämällä yhtenäistä nimeämiskäytäntöä, on mahdollista varmistaa, että kaikki projektia varten luodut assetit ovat helposti löydettävissä. Valmiit assetit on syytä nimetä selkeästi ja nimeämistapa pidettävä samanlaisena koko projektin

ajan. Projektityöskentelyssä nimeämiskäytännöistä on tärkeää sopia jo projektin alussa. Peliobjektin luomisessa voi olla monta eri vaihetta ja jokainen näistä tehtävistä voi kuulua eri henkilölle, joten on tärkeää, että muut artistit tietävät tarkalleen minkä assetin kanssa he työskentelevät. Nimeämiskäytäntöjen noudattaminen on myös tärkeää itsenäisessä työssä, sillä monet ohjelmat nimeävät samanlaiset objektit juoksevilla numeroinnilla. Oikean objektin löytäminen käyttöliittymästä voi olla vaikeaa, jos saman nimisiä objekteja on useita. (Chopine 2012, 17.)

Peligrafiikkaan käytettävien ohjelmistojen kanssa työskentely on yleensä raskasta tietokoneen suorituskyvylle ja jotkin toiminnot saattavat olla epävakaita, joten ohjelmat voivat kaatua aika ajoin. Tämän takia on suositeltavaa tallentaa työ tasaisin väliajoin. Useat ohjelmat osaavat tallentaa automaattisesti ja luoda varmuuskopioita säännöllisin väliajoin, joten näitä toimintoja kannattaa käyttää. Usean tunnin työn menettäminen ohjelman kaatuessa on turhauttavaa ja hidastaa työtä merkittävästi. Tallentaminen on tärkeää, kun työhön ollaan tekemässä merkittäviä muutoksia. (Chopine 2012, 17.) Uuden kopion tekeminen työtiedostosta on suositeltavaa, varsinkin silloin, kun ollaan tekemässä muutoksia, joita on vaikea peruuttaa jälkikäteen. Tällöin, jos jokin menee pieleen, on mahdollista palata aikaisempaan tiedostoon tekemättä peruttamattomia muutoksia.

3 PBR -STANDARDI

PBR, eli physically based rendering -standardi (suomeksi valo-oppiin perustuva renderöinti) on uusi varjostintekniikka, joka on kehittymässä nopeasti standardiksi peliteollisuudessa. Tietokoneiden kasvanut laskemisteho ja tarve standardisoida tekstuurigrafiikan tuottamista ovat johtaneet uuden varjostintekniikan kehittymiseen. PBR -tekniikka pyrkii uudelleenmäärittelemään, miten grafiikkaa luodaan ja renderöidään. (Wilson 2015.) Kuva 6 esittää Marmoset Toolbag 2 -ohjelmassa renderöityä kameran objektiivin 3D-mallia, jossa on käytetty PBR -standardilla toteutettuja tekstureita.



KUVA 6. Marmoset Toolbag 2 -ohjelmassa renderöity kameran objektiivi (Wilson 2015)

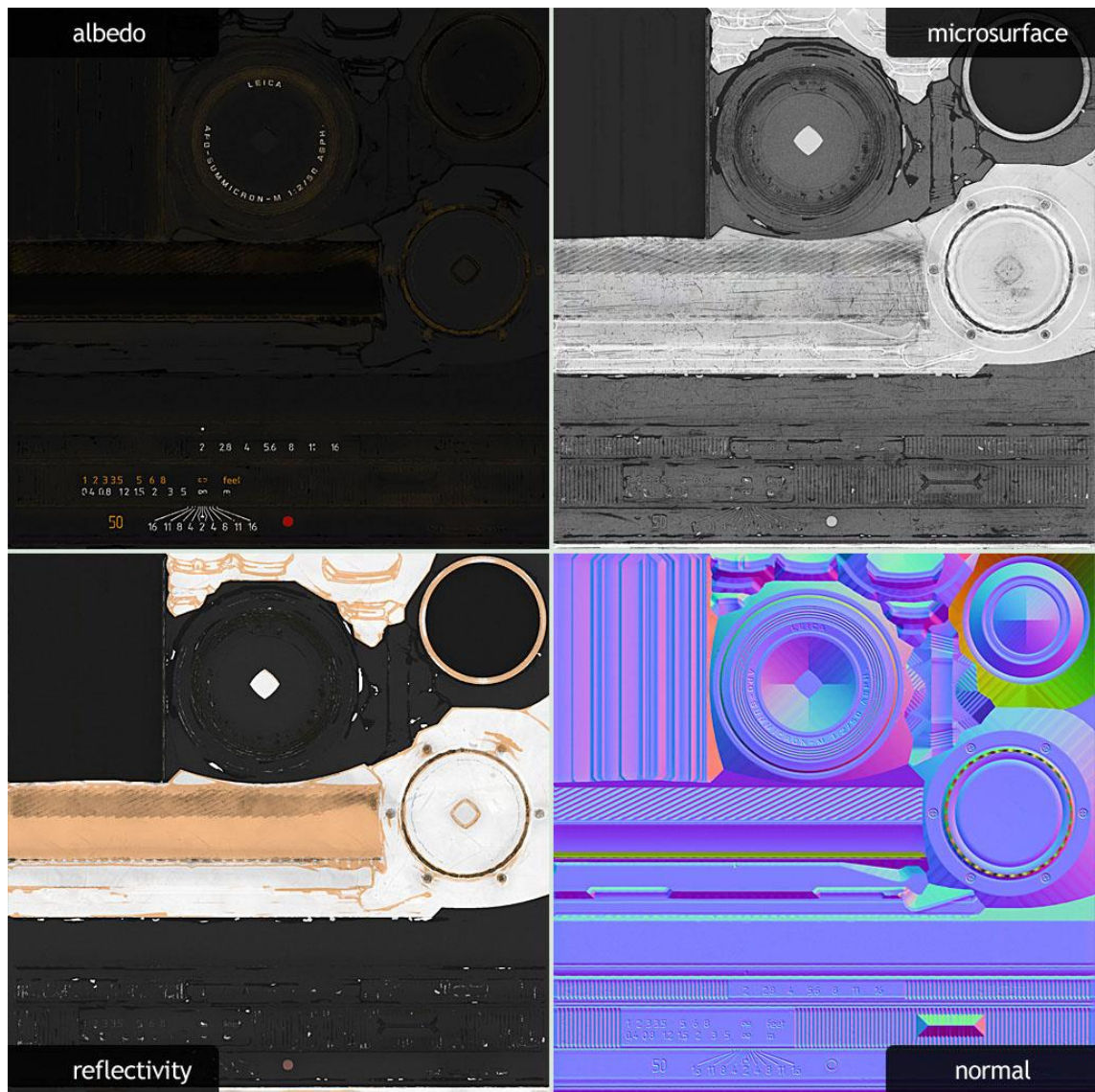
PBR perustuu konseptiin, jossa käytetään realistisia varjostus- ja valaistusmalleja mitattujen pinta-arvojen ohella, jotta oikean maailman materiaaleja voitaisiin jäljitellä mahdollisimman tarkasti. Oikeista materiaaleista mitattuja arvoja käytetään, koska renderöinnin laatua tärkeämpää on yhdenmukaisuus. Yhdenmukaisia arvoja käyttämällä artistit voivat jättää arvailun varaan perustuvan työskentelyn teksturoidessaan. Se myös takaa visuaalisen ohjauksen näkökulmasta sen, että artistitiimin tuottama grafiikka näyttää hyvältä jokaisessa mahdollisessa valaistusolosuhteessa. (Wilson 2015.)

Mikä tekee valo-oppiin perustuvasta varjostintekniikasta erilaisen verrattuna sitä edeltäviin tekniikoihin, on paljon yksityiskohtaisempi keskittyminen siihen, miten valo ja materiaalien pinnat käyttäytyvät. Varjostimien tarkkuus on kehittynyt tarpeeksi, jotta osa vanhoista arvioista valon käyttäytymisessä voidaan turvallisesti hylätä ja niiden mukana joitakin vanhempia tapoja grafiikan tuottamisesta. (Russel 2015.)

Vaikka teknikka onkin yleistymässä nopeasti, PBR ei välttämättä ole paras valinta jokaiseen projektiin. Jos peli ei pyri jäljittelemään tarkasti valon käyttäytymistä oikeassa maailmassa, tämän konseptin käyttäminen ei palvele projektia. Sen sijaan pelit, jotka pyrkivät realistiseen valaistukseen voivat kuitenkin hyötyä merkittävästi tästä tekniikasta. PBR -tekniikkaa käyttämällä voidaan saada ennustettavampia tuloksia eri valaistusolosuhteissa ja saada realistisia ja yhtenevämpiä tuloksia, koska kaikki perustuu oikeaan maailmaan. (Orsvärn 2015.)

PBR -standardin toteutus vaihtelee pelifirmojen sisällä, sillä se on enemmänkin konsepti kuin tiukka ohjenuora grafiikan tuottamiseen. Koska PBR-standardi perustuu kuitenkin samaan perusideaan, moni toteutustapa on helposti siirrettävissä projektista ja pelimoottorista toiseen. (Wilson 2015.)

Huolimatta PBR -standardin huomattavasti yleistyneestä käytöstä, Rescue 2: Everyday Heroes -peli ei hyödyntänyt uutta tekniikkaa. PBR -teknologiaa tukevat varjostimet esiteltiin projektissa käytettyyn Unity-pelimoottoriin vasta projektin loppupuolella, joten standardia ei pystytty ottamaan käyttöön sen perinteisestä teksturointitavasta melko huomattavasti eroavan luomisprosessin takia. Pelin realismia hakevan luonteen takia PBR -tekniikka olisi kuitenkin sopinut pelin graafiseen ilmeeseen. Kuva 7 esittää neljää kameran objektiivin käyttämää PBR-standardin mukaista tekstuurikarttaa ja kuva 8 kolmea perinteisen varjostintekniikan käyttämää tekstuuria. Perinteiseen tapaan verrattuna PBR -standardia käyttävissä tekstuureissa valo-informaatio on poistettu muista kartoista ja se on sisällytetty erilliseen tekstuuriin.



KUVA 7. Kameran objektin käyttämät PBR-standardin mukaiset tekstuurikartat (Wilson 2015)



KUVA 8. Perinteisen varjostintekniikan käyttämät tekstuurikartat (Wilson 2015)

4 MITÄ OVAT TEKSTUURIKARTAT?

Tekstuurikartat ovat kaksiulotteisia bittikarttakuvia, joita käytetään 3D-mallien pinnoilla. Objektin pohjaväriin lisäksi tekstuurikartat voivat esittää monia muita objektin pintarakenteen ominaisuuksia, kuten heijastavuutta tai karheutta. Tekstuurikartat eroavat ulkonäöltään toisistaan sen mukaan, mitä fysikaalista ominaisuutta ne kuvastavat. (Unity Manual 2017.)

Tekstuurikarttojen avulla on mahdollista parantaa tietokoneen suorituskykyä reaaliaikaista kuvaa laskettaessa, kun pinnanmuotojen jäljittelemiseen ei tarvitse käyttää varsinaista geometriaa. Nykyään 3D-mallien pinnoilla käytetään useita erilaisia tekstuurikarttoja, jotka yhdessä pyrkivät luomaan mahdollisimman realistisen lopputuloksen. (Computer Graphics Wiki 2017.)

Kun tekstuurikarttoja luodaan, artistilla on aina tiettyjä vaatimuksia siitä, millainen tekstuurin kuuluu olla. Näitä ovat mitat sekä erilaiset rajoitukset ja säännöt, joita tulee noudattaa. Projekti, jossa artisti työskentelee, käytettävä teknologia ja pelin lajityyppi vaikuttavat kaikki tekstuurien laatuvaatimukseen ja ohjelinjoihin. (Gahan 2009 61, 63.) Vaikka teksturoiminen on hyvin visuaalista työtä, artistin on myös tärkeä tuntea prosessin osien tekninen puoli käyttääkseen tekstureita oikeaoppisesti ja tehokkaasti. Tietämättömyys konseptien toimintatavasta johtaa vääriin tottumuksiin ja epäammattimaiseen lopputulokseen.

Tekstuurien resoluutiolle ei ole tarkkoja sääntöjä, sillä pelimoottorit pystyvät käsittelemään minkä kokoisia bittikarttakuvia tahansa. Peliteollisuudessa käytäntönä on kuitenkin käyttää neliön tai suorakaiteen muotoisia, kooltaan kahden potenssia käyttäviä kuvatiedostoja. Useimmat näytönohjaimet ja pelimoottorit käyttävät kahden potenssia, sillä ne voivat käsitellä ja piirtää tekstureita nopeammin tällä tavoin. Peliprojekteissa pääartistien tekemät linjaukset usein määrittelevät, minkä kokoisia tekstuurikartoista pitää tehdä. Yleisimmät käytetyt koot ovat 256 x 256, 512 x 512 ja 1024 x 1024 pikseliä. Isompi resoluutio mahdollistaa yksityiskohtien tarkemman esittämisen ja vähentää rakeisuutta, mutta samalla nostaa tiedostokokoa ja laskentatehon vaatimuksia. Korkeaan laatuun pyrkivät pelit käyttävät nykyään vähintään 2048 x 2048 pikselin kokoisia tekstuuritiedostoja. Laatoittuvissa tekstureissa neliön muotoiset tekstuurikartat toimivat

parhaiten niiden helpon toistettavuuden takia. Varsinkin pelimaailmaa koostettaessa yhdenmukaiset rakennuspalaset helpottavat maailman rakentamista. (Gahan 2009, 63, 65, 67.)

Tekstuurikartoissa voidaan käyttää erilaisia tiedostomuotoja. Projektin vaatimukset vaikuttavat projektissa käytettäviin tiedostomuotoihin. Tiedostomuodon valintaan vaikuttavat tiedoston tarkkuus, sen erilaiset ominaisuudet sekä pelimoottorin tekniset rajoitukset ja vaatimukset. Tekstuurikarttojen luomisprosessin aikana käytetään yleisesti eri tiedostotyyppiä kuin pelimoottoriin vietävässä valmiissa assetissa. Esimerkiksi Photoshop-kuvankäsittelyohjelman PSD-tiedostomuoto on teksturointiprosessissa joustava, koska se pystyy tallentamaan paljon erilaista tietoa useille tasoille, joita voidaan tarvittaessa muokata erikseen. Korkean muokkauskyvyn sisältävät tiedostot ovat tiedostokooltaan kuitenkin hyvin isoja eikä niitä ole käytännöllistä jättää valmiiseen projektiin. Tämän takia valmis tekstuurikartta pakataan toiseen muotoon tiedostokoon minimoimiseksi. Kuvatiedostojen pakkaustavat vaihtelevat keskenään, ja mitä enemmän kuvatiedostoa pakataan, sitä enemmän kuvan laatu kärsii. (Gahan 2009, 50-52, 60.)

5 TEKSTUURIKARTTATYYPIT

Jotta graafikko saa pelimootorissa käytettävistä kolmiulotteisista objekteista realistisen näköisiä, niiden pinnalle heijastetaan yleensä useampia erilaisia bittikarttoja, jotka vaikuttavat pinnan ulkonäköön. Jokaisella tekstuurikarttatyypillä on oma tehtävänsä pinnan eri ominaisuuksien kuvantamisessa. Eri tekstuurikarttojen käyttäminen riippuu siitä, millainen pelin tyylistä halutaan, mitä teknologiaa pelimootori käyttää ja mille alustalle peliä kehitetään. Yleisimpiä pelikehityksessä käytettyjä tekstuurikarttatyyppejä ovat muun muassa diffuusio-, spekulaarisuus- ja normaalikartta.

5.1 Diffuusiokartta

Diffuusiokartalla (engl. diffuse map) tarkoitetaan yleensä värikarttaa, joka sisältää objektin pinnan väri-informaation. Nykypäivän peligrafiikassa diffuusiokartta sisältää pääasiassa pinnan väri-informaation, mutta poikkeuksia tähänkin voidaan tehdä. Koska pelimootorit eivät vielä nykyäänkään jäljennä oikeaa maailmaa täysin tarkasti, on diffuusiokarttaan mahdollista lisätä hienovaraisia yksityiskohtia kertomaan valoinformaatiota objektin pinnasta. Metallipaneelien ja puulankkujen välit ovat esimerkkejä, joissa varjoja voidaan maalata värikarttaan lisäämään aitouden tuntua. Myös halkeamat ja saumat ovat paikkoja, joihin kasaantuu yleensä likaa ja jotka lisäävät alueiden tummuutta tekstuurissa. Näihin on mahdollista vaikuttaa muillakin tekstuurikartoilla, mutta liian muovisen ulkonäön välttämiseksi lisäinformaation maalaaminen suoraan diffusioteksturiin voi olla suositeltavaa. (Ahearn 2008, 74.)

Kuva 9 esittää Rescue 2 -pelissä käytetyn rakennuksen diffuusiokarttaa. Tekstuurissa on nähtävissä väri-informaation lisäksi myös valo- ja varjoinformaatiota.



KUVA 9. Rescue 2 -pelissä käytetyn rakennuksen diffuusiokartta.

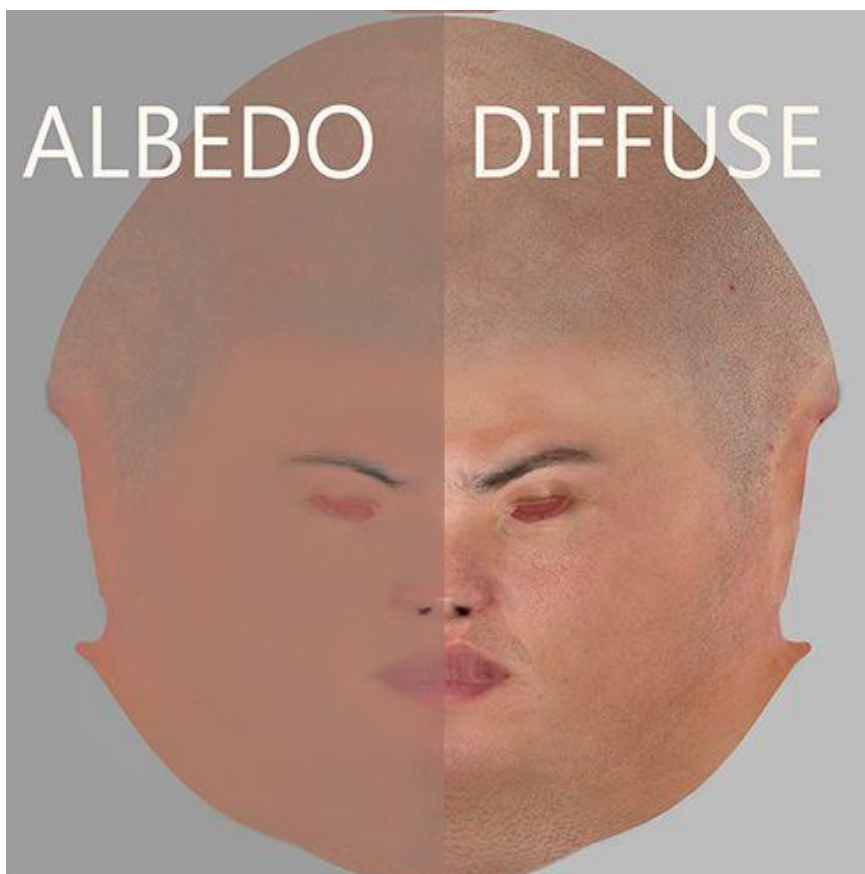
Teksturoidessaan pelimalleja artisteilla oli alun perin käytettävissään ainoastaan diffuusiokanava, joten varjot ja valaistukset maalattiin diffuusiotekstuuriin itseensä. Koska valoinformaatio oli staattinen, se ei reagoinut pelimoottorin valaistuksen kanssa. Tämä työnkulku on viime vuosina muuttunut, kun uudet teknologiat ovat jakaneet tekstuureissa välitettävän tiedon useampiin eri komponentteihin, jotka kootaan yhteen lopputuloksen aikaansaamiseksi. Näin värikartasta on tullut huomattavasti yksinkertaisempi muun kuin väri-informaation kannalta. (Ahearn 2008, 74.)

Rescue 2 -pelin graafinen ulkoasu hyödynsi suurelta osin pelkkien diffuusiokarttojen käyttöä, jolloin valojen ja varjojen lisääminen karttaan oli pakollista yhtenäisen ulkonäön aikaansaamiseksi. Tämä johtui ajankäytön minimoimisen lisäksi pääasiassa pelin graafikoiden valinnasta teksturoinnin työnkulussa ja siitä, että projektissa käytettiin varjostimia, mitkä eivät hyödyntäneet uudempaa kuvantamisstandardia.

5.2 Perusvärikartta

Perusvärikartta (engl. base color map) määrittää objektin pinnasta sironneen valon värin. Se on uudemman varjostintekniikan käyttämä versio perinteisestä diffuusiokartasta, joka voi sisältää tietoa sekä heijastuvan valon väristä epämetalleissa, että heijastuskykyarvon määrästä metalleissa. Perusvärikartta voidaan mieltää jotakuinkin monotoniseksi

sen värienkäytön kannalta, koska siinä on vähemmän kontrastia kuin perinteisessä diffuusiokartassa. Perusvärikartassa olevien värien ei siten tulisi olla liian kirkkaita tai tummia ja niissä käytetyt värit tulisi pitää tietyn kirkkausarvojoukon sisällä. (Wilson 2015.) Kuva 10 havainnollistaa diffuusio- ja perusvärikartan eroja ihmiskasvojen tekstuurissa. Vasemmanpuoleisesta tekstuurista on riisuttu valoinformaatio, jolloin se näyttää ilmeettömältä.



KUVA 10. Diffuusio- ja perusvärikartan vertailukuva (Daz 3D Forums 2015)

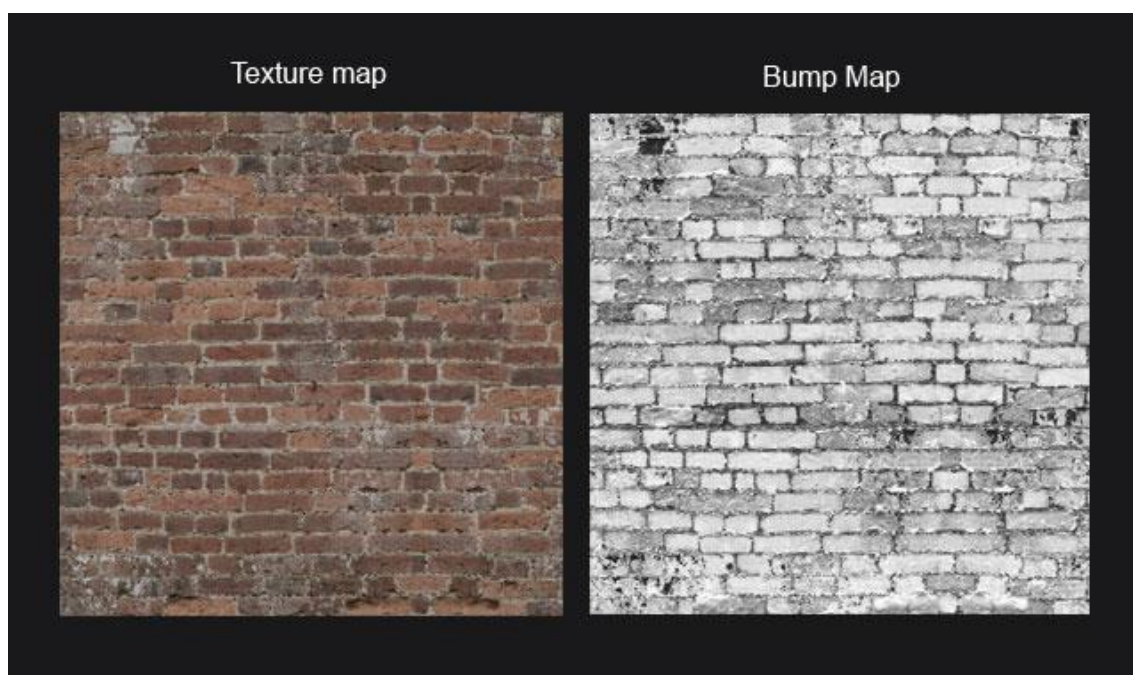
Koska perusvärikarttaan tallennetaan tietoa heijastuvasta valosta, sen ei tulisi sisältää valoinformaatiota, kuten ympäristön sulkeumaa. Poikkeuksia voidaan tehdä mikrotason okkluusiossa, jota varjostin ei muuten pystyisi esittämään pelkkää ympäristön okkluusiokarttaa käyttämällä. Kuitenkin, jos mikro-okkluusio lisätään perusvärikarttaan, siinä olevien väriarvojen tulee silti pysyä määriteltyjen kirkkausarvojen sisällä. (McDermott 2017, 5-6.)

Riippuen käytetystä ohjelmistosta ja työtavasta, perusvärikarttaa voidaan kutsua myös nimellä albedo-kartta (engl. albedo map). (Wilson 2015.)

5.3 Kohoumakartta

Kohoumakartta (engl. bump map) on yksi vanhempia karttoja, jota nykypäivänä käytetään. Kohoumakartat luovat syvyysvaikutelman kolmiulotteisen mallin pintaan käyttäen yksinkertaista valaistustemppua, joten kartta ei lisää ylimääräistä geometriaa mallin pintaan. (Pluralsight 2014.)

Tyypillisesti kohoumakartat ovat harmaasävyisiä bittikarttakuvia. Näitä harmaasävyjen arvoja käytetään kertomaan 3D-ohjelmalle kahta asiaa: onko pinta ylempänä vai alempana geometrian pinnasta. Kohoumakartassa lähellä keskiharmaata olevat arvot kuvantuvat pelimoottorissa heikkoina yksityiskohtina objektin pintaan. Arvojen mennessä vaaleammiksi, yksityiskohdat näyttävät työntyvät ulos objektin pinnasta. Vastaavasti arvojen mennessä tummemmiksi ja lähelle mustaa, yksityiskohdat näyttävät painuvan pinnan sisään. (Pluralsight 2014.) Kuva 11 esittää tiiliseinän diffuusio- ja kohoumakarttaa. Vasemmanpuoleista diffuusiokarttaa käyttämällä on luotu sitä vastaava kohoumakartta.



KUVA 11. Diffuusio- ja kohoumakartta (3D Modeling 4 Business 2012)

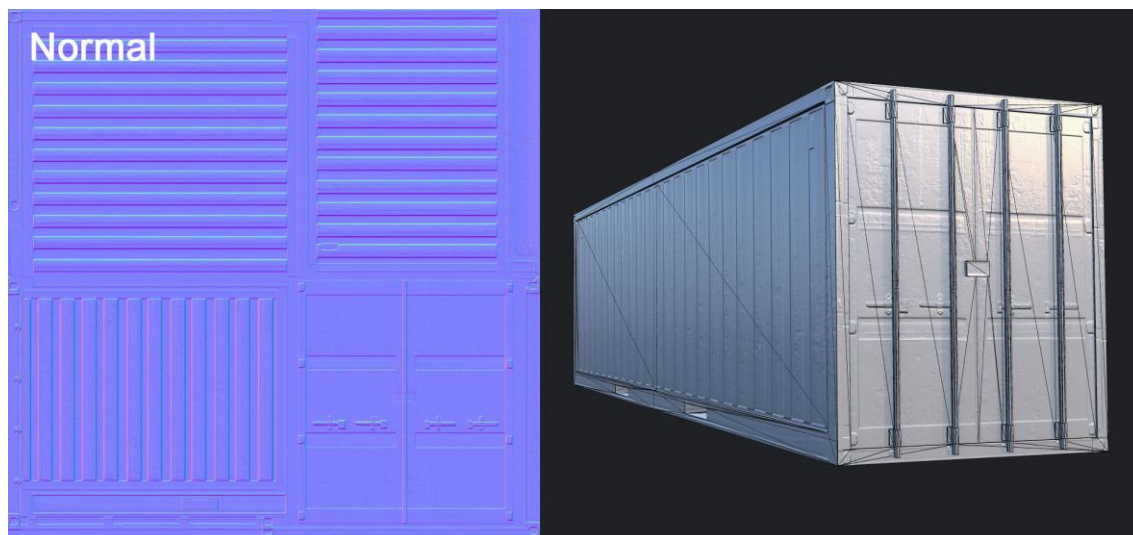
Kohoumakartat ovat mainioita luomaan pieniä yksityiskohtia kolmiulotteisen malliin. Niitä on myös verrattain helppo luoda ja muokata kaksikulotteisissa kuvankäsittelyohjelmissa, kuten Photoshopissa, koska käytössä on ainoastaan harmaasävyjä. Huonona

puolena kohoumakartoissa on, että niiden luoma syvyysvaikutelma rikkuu helposti, jos niitä tarkastellaan väärästä kuvakulmasta. On myös tärkeää huomata, että koska kohoumakartta ei lisää geometriaa objektin pinnassa, pysyy objektin tahkojen siluetti muuttumattomana kartasta huolimatta. (Pluralsight 2014.)

5.4 Normaalikartta

Normaalikartta (engl. normal map) nimitys viittaa uudempaan ja parempaan versioon vanhemmasta kohoumakartasta. Kohoumakarttojen tavoin niiden lisäämät yksityiskohdat ovat illuusiota, eivätkä oikeasti lisää geometriaa mallin pintaan. Normaalikartta luo illuusion yksityiskohtien syvyydestä mallin pinnalle, mutta tekee sen eri tavalla kuin kohoumakartta. (Pluralsight 2014.)

Kuva 12 esittää Rescue 2 -pelissä käytetyn satamakontin normaalikarttaa ja sen 3D-mallia. Normaalikartan avulla 3D-mallin kyljissä ja päädyssä olevat pinnanmuodot on saavutettu ilman ylimääräistä geometriaa.



KUVA 12. Rescue 2 -pelissä käytetty satamakontti ja sen normaalikartta.

Siinä missä kohoumakartta käyttää harmaasävyjen arvoja luodakseen korkeus- tai syvyysvaikutelman, normaalikartta käyttää RGB-kanavien informaatiota, joka vastaa suoraan X-, Y- ja Z-akseleita 3D-avaruudessa. Tämä informaatio kertoo 3D-ohjelmalle,

miten normaalit ovat suuntautuneet jokaisessa mallin tahkossa. Näin ohjelma osaa varjostaa mallin tahkot reaaliajassa. (Pluralsight 2014.)

Normaalikartoista puhuttaessa on tärkeää huomata, että niistä on olemassa kahta täysin erilaista tyyppiä, joiden ulkoasu vaihtelee huomattavasti, kun niitä tarkastellaan kaksiulotteisessa muodossa. Yleisin käytetty versio on tangettiavaruuksnormaalikartta (engl. Tangent Space normal map), joka on ulkoasultaan sekoitus pääasiassa purppuran ja sinisen sävyjä. Tangettiavaruuksnormaalikartta toimii parhaiten geometriassa, jonka täytyy deformoitua animoitaessa. Tämän takia tangentiavaruuksnormaalikarttaa onkin hyvä käyttää esimerkiksi pelihahmoissa. (Pluralsight. 2014.)

Malleissa, joiden ei tarvitse deformoitua voidaan käyttää objektiavaruuksnormaalikarttaa (engl. Object Space normal map). Nämä kartat käyttävät tangettiavaruuksnormaalikarttoja huomattavasti enemmän värejä ja niiden suorituskyky on hieman parempi edellisiin verrattuna. Lukuisat värit tekevät karttojen muokkaamisesta tai luomisesta kaksiulotteisissa kuvankäsittelyohjelmissa kuitenkin erittäin hankalaa. (Pluralsight 2014.)

Normaalikartan yleisin käyttötapana on peittää yksityiskohtaisen, korkearesoluutioisen mallin ja low-polygon -mallin eroavaisuudet. Normaalikarttoja voidaan luoda ”leipomalla” yksityiskohdat high-poly -mallista bittikarttaan, jota yksinkertaisempi low-poly -malli käyttää. Normaalikarttojen käyttäminen mahdollistaa matalaresoluutioisten mallien käyttämisen videopeleissä antaen niille silti vaikutelman korkeammasta yksityiskohtien tasosta. (Birn 2013, 337-338.)

Rescue 2 -pelissä normaalikarttoja käytettiin usein peliobjekteissa, joiden haluttiin olevan erityisen yksityiskohtaisia, kuten ajoneuvoissa ja pelaajan ohjailemissa pelihahmoissa. Näiden lisäksi kameran lähelle tulevat elementit, kuten rakennusten katot käyttivät normaalikarttoja, jotta diffuusiokarttojen luomaa aitouden tuntua saatiin syvennettyä ilman ylimääräistä geometriaa.

Ajankäytön säästämiseksi korkearesoluutioisia 3D-malleja, joista normaalikartat saataisiin, ei juurikaan tehty, vaan niiden luomiseen käytettiin vaihtoehtoisia menetelmiä. Luominen tapahtui suureksi osaksi generoimalla erilaisista valokuvista objektien pinnoille sopivia kohoumia ja syvennyksiä sekä maalaamalla korkeuskuvioita itse ja muut-

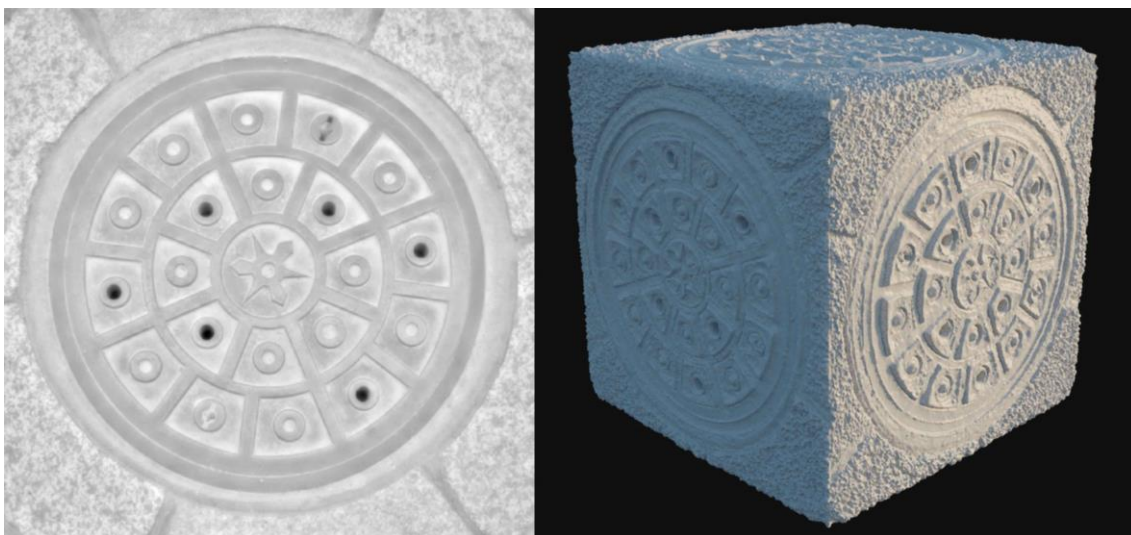
taen ne normaalikartan muotoon käyttämällä erityistä kuvankäsittelyohjelman liitännäistä.

5.5 Siirtymäkartta

Siirtymäkarttoja (engl. displacement map) käytetään kohouma- ja normaalikarttojen tavoin luomaan yksityiskohtia objektien pinnoille. Kuten kartan nimi antaa ymmärtää, siirtymäkartta siirtää fyysisesti geometriaa objektissa, joka käyttää sitä (Pluralsight 2014). Siinä missä kohouma- ja normaalikartat vääristävät varjostusta kohoumien tai painautumien illuusion luomiseksi, siirtymäkartat luovat oikeita kuhmuja, ryppyjä ynnä muita varsinaiseen geometriaan. Täten pinnanmuodot voivat muodostaa varjoja ja peittää muita objekteja, mutta vaativat huomattavasti enemmän avaruuspisteitä geometrias-
sa toimiakseen. Jotta siirtymäkartan mukaisia yksityiskohtia voidaan luoda, objektin geometriaa täytyy jakaa pienempiin osiin, jotta geometriaa on mahdollista muokata. (Blender 2.78 Manual 2017.)

Siirtymäkartat ovat käytännöllisiä siitä, että niitä on mahdollista luoda joko ns. ”leipomalla” bittikarttakuva korkearesoluutioisesta mallista tai kuvankäsittelyohjelmassa maa-
laamalla. Kohoumakartan tavoin siirtymäkartta käyttää harmaasävyjä korkeusinformaation välittämiseen. Huonona puolena siirtymäkartan vaatima lisägeometria vaatii mallin-
tajalta lisätyötä ja kasvattaa aikaa kuvan piirtämisessä. Tämän takia moni 3D-ohjelmisto laskee siirtymäkartan muutokset geometriaan vasta kuvaa piirrettäessä. (Pluralsight 2014.)

Liian vähäisen geometrian käyttäminen saattaa johtaa rosoiseen tai huonosti tarkennettuun lopputulokseen objektin pinnassa, missä kartta vaikuttaa. (Birn 2013, 334.) Kohouma- ja normaalikarttoihin verrattaessa piirtämiseen käytetty aika on huomattavasti isompi, mutta lisägeometrian ansiosta siirtymäkartan lopputulokset ovat merkittävästi paremmat. Artistin tulisi aina punnita siirtymäkartan tuomat hyödyt verrattuna sen huonoihin puoliin ennen niiden käyttämisen päättämisestä. (Pluralsight 2014.) Kuvassa 13 nähdään siirtymäkartta ja renderöity kuva kuutiosta, jonka päälle siirtymäkartta on levitetty.



KUVA 13. Siirtymäkartta ja sitä käyttävä kuutio (Lampel 2017)

5.6 Spekulaarisuuskartta

Spekulaarisuuskartta (engl. specular map) vaihtelee spekulaarisuusheijastuksen kirkkautta ja väriä eri puolilla mallin pintaa. Spekulaarisuuskartta ei itsessään luo heijastuksia, vaan heijastusten on tultava ulkopuolisista valolähteistä. Spekulaarisuuskartta voi sävyttää heijastuksia, muuttaa niiden kirkkautta tai jopa estää niiden muodostumisen kokonaan halutussa paikassa mallin pinnalla. (Ahearn 2008, 80.)

Spekulaarisuuskartan aikaansaamat lopputulokset ovat näkyviä ainoastaan kohdissa, jossa heijastus olisi muutenkin mahdollinen. (Birn 2013, 328-329.) Spekulaarisuuskarttoja voi olla sekä harmaasävyisiä, että RGB-väriarvoja käyttäviä. Luonnollisesti harmaasävyinen spekulaarisuuskartta sävyttää heijastuksia ainoastaan saapuvan valon värin perusteella, värillinen spekulaarisuuskartta taas sävyttää heijastuksia bittikarttaan määriteltujen väriarvojen perusteella. (Ahearn 2008, 80.)

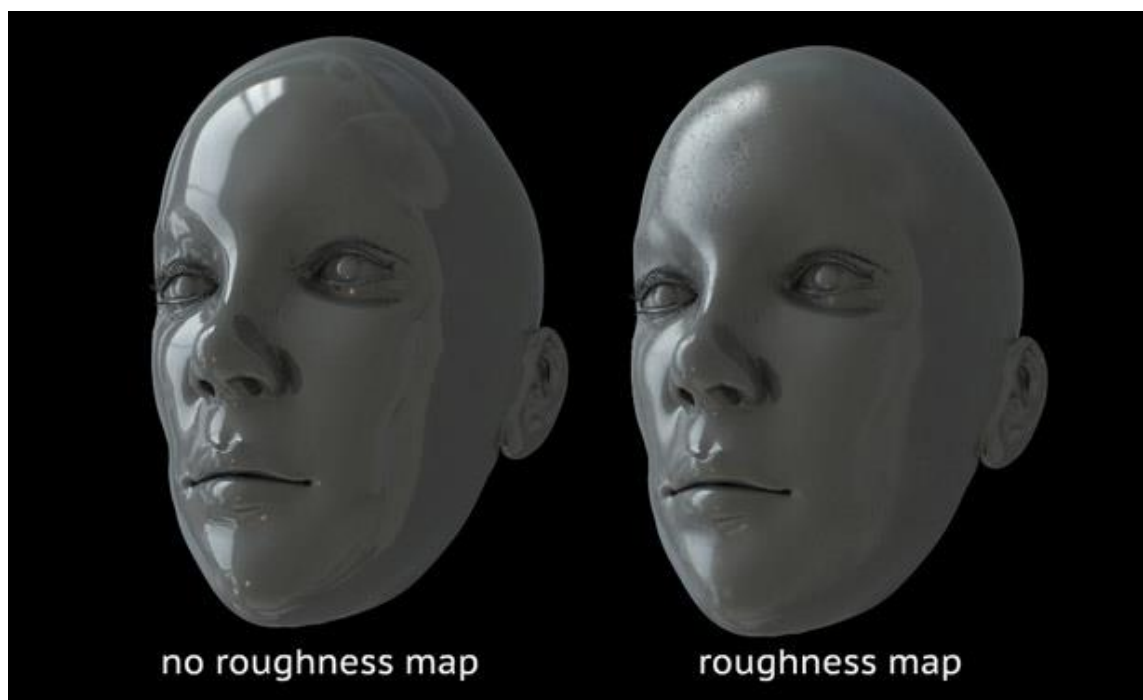
Kirkkaat alueet spekulaarisuuskartassa tekevät heijastuksista kirkkaampia, luoden kirkkaamman alueen objektin pinnalle. Tummat sävyt kartassa taas tekevät heijastuksista vähemmän näkyviä ja täysin mustat alueet estävät heijastuksien muodostumisen vastavissa alueissa kokonaan. (Birn 2013, 328-329.)

5.7 Kiilto-/karheuskartta

Kiilto- ja karheuskartta (engl. gloss/roughness map) kuvastavat objektin pinnan epätasaisuuksia, jotka aiheuttavan valon hajaantumisen sen osuessa pintaan (McDermott 2017, 11). Ne määrittävät, kuinka terävä tai karhea materiaalin pinnan peiliheijastus on, eli kuinka laaja tai kapea heijastus on (Wilson 2014).

Heijastuneen valon suunta vaihtelee satunnaisesti pinnan epätasaisuuden mukaan, mutta valon intensiteetti pysyy muuttumattomana. Karheammissa pinnoissa heijastukset näkyvät isompina ja himmeämpinä. Sileämmät pinnat taas säilyttävät peiliheijastuksen terävämpänä, joka saattaa näyttää kirkkaammalta vaikkakin heijastuneen valon määrä on sama. (McDermott 2017, 11.)

Kuva 15 esittää kahta ihmishahmon pään 3D-mallia, joista toisessa on käytetty karheuskarttaa. Karheuskarttaa käyttävän 3D-mallin heijastukset ovat sameampia, kuin mallin, jossa ei ole karttaa, vaikka valaistus on sama.



KUVA 15. Kaksi ihmishahmon päätä esittävää 3D-mallia, joista toinen käyttää karheuskarttaa (UE4 AnswerHub 2015)

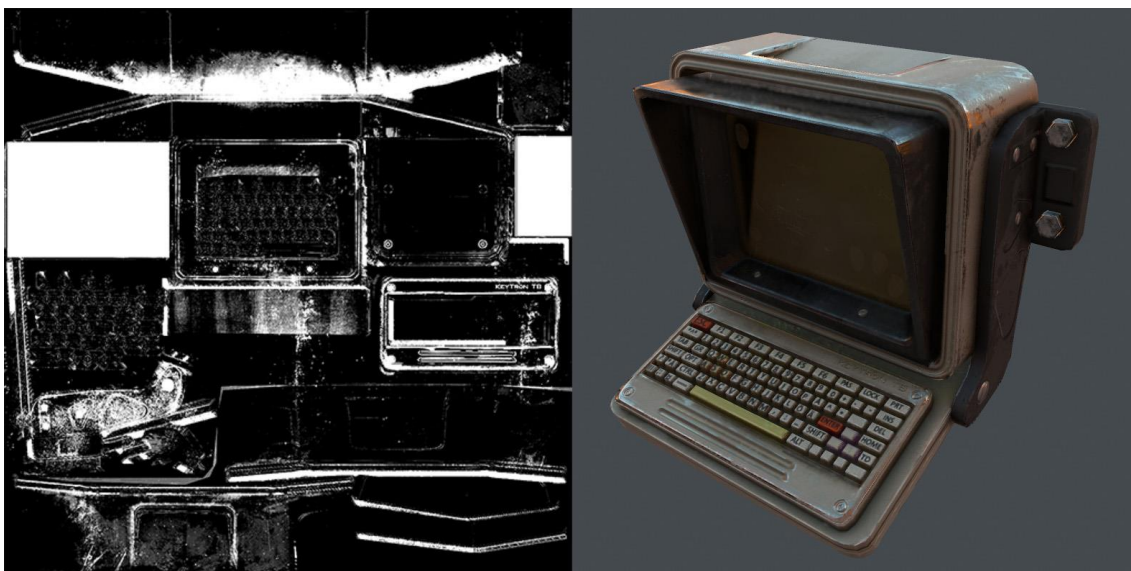
Kiilto- ja karheuskartassa pinnan karheus- tai sileysinformaatio ilmoitetaan harmaasävyinä. Kiiltokartassa tummia arvoja käytetään määrittelemään mattapintaisia materiaaleja, kun taas vaaleammat arvot kuvastavat kiiltäväpintaisia materiaaleja. (Wilson 2014.) Karheuskartta kertoo saman informaation kuin kiiltokarttakin, mutta käänteisesti. Karheuskartassa musta esittää sileää pintaa ja valkoinen karheaa. (McDermott 2017, 11.)

Kiilto- ja karheuskartta ovat luovimpia karttoja, sillä ne antavat artistille mahdollisuuden määrittää pinnan luonteen visuaalisesti. Niillä artisti voi kertoa tarinan pinnan materiaalin kunnosta, millaisessa ympäristössä se on ollut, onko sitä käsitelty huolellisesti vai huolimattomasti ja millaisille luonnonvoimille se on altistunut. Objektin pinnan kunto kertoo paljon sitä ympäröivästä maailmasta ja on näin ollen iso tekijä objektien ja ympäristöjen suunnittelussa, mitä artisti koittaa luoda. (McDermott 2017, 11.)

5.8 Metallikartta

Metallikarttaa (engl. metallic map) käytetään määrittelemään, mitkä alueet materiaalista luetaan puhtaina metalleina. Metallikartta on harmaasävyinen bittikartta, joka käyttäytyy samankaltaisesti kuin peitemaski, sillä se kertoo varjostimelle, miten sen kuuluu tulkita dataa, joka on esitetty perusvärikartassa. (McDermott 2017, 11.)

Kuva 16 esittää tietokonepäänteen metallikarttaa ja renderöityä 3D-mallia. Renderöidystä kuvasta voidaan nähdä maalin alta paljastuva metalli, joka kiiltää maalia huomattavasti enemmän. Paljastunut metalli on värikoodattu metallikarttaan valkoisella.



KUVA 16. Tietokonepääteen metallikartta ja renderöity lopputulos (Sketchfab Blog 2015)

Metallikartta määrittelee varjostimelle, mitkä alueet perusvärikartassa tulisi tulkita eristeiden heijastusväriarvoina ja mitkä alueet metallin heijastekykyarvoina. Metallikartta ei siis sisällä dataa, jota käytettäisiin suoraan materiaalin arvona. Käytännössä, kun varjostin katsoo metallikarttaa ja näkee valkoista, se tarkastaa vastaavat alueet perusvärikartassa saadakseen heijastuskykyarvot metallille. Metallikartassa musta väri edustaa epämetallia ja valkoinen väri puhdasta metallia. Metallien ja epämetallien määrittelyn suhteen metallikartta on usein kaksiarvoinen, eli jos bittikartan väri on musta tai valkoinen, niin materiaali on metalli tai epämetalli. (McDermott 2017, 11.)

Rapautuneita pintoja jäljennettäessä metallia voidaan käsitellä hapettuneena tai korvata sitä muilla ympäristöllisillä jäljillä, kuten tahroilla ja pinttyneellä lialla. Hapettunutta metallia täytyy käsitellä eristeenä, esimerkiksi ruosteena. Sama pätee maalatun metallin kanssa. Katsottaessa maalattua metallipintaa, missä osa maalista on lohkeillut pois, paljastunut metallipinta on puhdasta metallia ja maalipinta taas eristeinen kerros. (McDermott 2017, 11.)

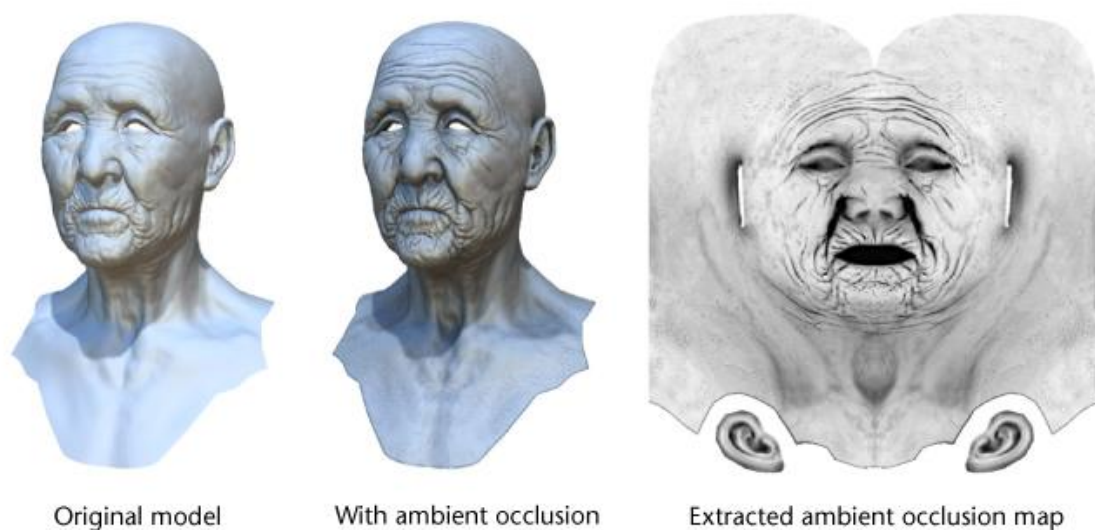
Vaikka metallikartta onkin useasti mustavalkoinen, voidaan siinä käyttää myös harmaasävyjä. Metallikartta voi esimerkiksi kuvata sekoittunutta muotoa metallin ja epämetallin välillä. Tällaisia kohtia esitetään kartassa harmaasävyillä. Esimerkkinä voidaan ottaa objekti, jonka pinnassa on likakerros, joka osittain peittää metallia. Lika on eriste

ja jos nämä kohdat jättää täysin valkoiseksi metallikartassa, perusvärikartassa kuvattuja tahroja käsiteltäisiin metallin heijastuskykyarvoilla. Laskemalla metallikartan valkoisuutta kohdissa, joissa likaa esiintyy, on mahdollista luoda sopiva sekoitus eristeen ja metallin heijastuskykyarvoja. (McDermott 2017, 11)

5.9 Ympäristön okkluusiokartta

Ympäristön okkluusiokarttaa (engl. ambient occlusion map tai lyhyemmin AO -map) käytetään simuloimaan pehmeää, globaalia varjostusta ympäristössä. Okkluusiokartta luodaan 3D-mallinnusohjelmassa hyödyntäen objektien omaa geometriaa ja niiden ympäristöä. (Birn 2014, 89.)

Kuva 17 esittää ympäristön okkluusiokarttaa sekä kahta ihmishahmon 3D-mallia, joista oikeanpuolimmaisen päälle on levitetty AO-kartta. Kuvasta voidaan huomata, että karttaa käyttävän mallin kasvonpiirteet erottuvat selvemmin, kun niihin on simuloitu varjostuksia.



KUVA 17. Ympäristön okkluusiokartta sekä kaksi 3D-mallia, joista toinen käyttää AO-karttaa (Autodesk Mudbox Online Help 2010)

Okkluusiota simuloitaessa objektit ympäröidään kirkkaalla valkoisella valolla, ilman suoraa valonlähdetä. Pinnat, jotka altistuvat valolle täysin, eivät varjostu ollenkaan ja ovat näin ollen kirkkaita. Muut pinnat, kuten esimerkiksi rakennusten välissä olevat kujat tai objektien väliin jäävät aukot ovat osittain piilossa valolta, joten ne piirtyvät

tummempina. Esineiden alla, nurkissa ja syvimmissä halkeamissa pinnat, jotka ovat täysin peitossa valolta, piirtyvät mustina. (Birn 2014, 89.) Ympäristön sulkeumaa ei tapahdu oikeassa elämässä eikä se perustu fysikaalisesti oikeaoppiseen valaistukseen. (Blender 2.78 Manual 2017.)

Simuloinnissa varjostuneet alueet leivotaan geometrian ja UV-kartoituksen perusteella kaksiulotteiseen bittikarttaan, jota on mahdollista soveltaa erilaisilla tavoilla. Vanhemmassa teksturointitavassa okkluusiokartta on tapana sekoittaa diffuusio- ja spekulariisuuskarttaan näyttävämmän varjostuksen saavuttamiseksi ja muistin säästämiseksi. Huonona puolena tässä on, että diffuusio- ja spekulariisuuskarttaan sekoitettu varjostus ei reagoi dynaamiseen valaistukseen. (Polycount 2010.) Spekulariisuuskarttaan lisätty okkluusio tuottaa virheellisiä heijastuksia, koska varjostuneet alueet eivät ole sama asia kuin vähemmän heijastava pinta.

Koska okkluusiokartta ei perustu realistiseen valaistukseen, artisti voi joutua myös korjailemaan okkluusiokartan varjostuksia liiallisten valaistusvirheiden välttämiseksi. (Polycount Wiki 2015.) Uutta tekniikkaa hyödyntävät varjostimet voivat sisältää kanavan okkluusiokartalle, mihin okkluusiokartan voi laittaa ilman sen sekoittamista muihin karttatyypeihin. Tällöin valon osuessa objektin pintaan varjostin osaa häivyttää karttaan leivotut varjot. Erillisen kartan käyttäminen tosin vie enemmän muistia. (Polycount 2010.)

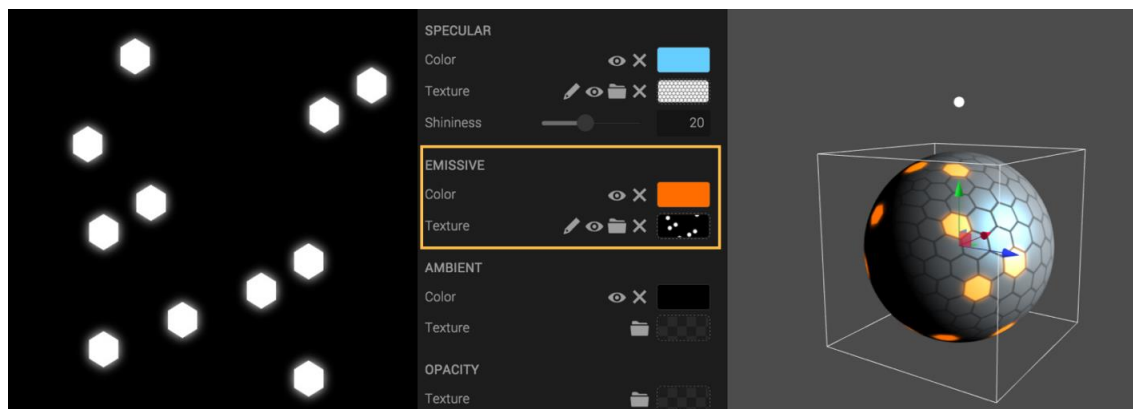
Ympäristön okkluusiota on myös mahdollista simuloida pelimoottorin sisällä reaaliajassa ilman erillistä tekstuuria. Tätä tekniikkaa kutsutaan nimellä Screen Space Ambient Occlusion (lyh. SSAO). Nykypäivänä tekniikka on laajalti käytössä peliteollisuudessa. Vaikka erillisiä leivottuja tekstuureita ei tarvita, tekniikka tarvitsee hyvän näyttökortin ja se tuottaa jonkin verran varjostusvirheitä. (Polycount Wiki 2015.)

SSAO -tekniikkaa voidaan hyödyntää peleissä muiden okkluusiotekniikoiden ohella (Polycount 2010). Myös 3D-mallien avaruuspisteisiin on mahdollista tallentaa tietoa valaistuksesta, jolloin avaruuspisteet varjostavat tahkot leivotun okkluusion mukaan. Tämä tekniikka säästää muistia, mutta varjostuksen laatu on riippuvainen geometrian tarkkuudesta. Harvempi avaruuspisteiden määrä tuottaa epätarkan varjostuksen, suuri vastaavasti tarkemman. (Polycount Wiki 2015.)

Ympäristön okklusiokarttaa hyödynnettiin Rescue 2-projektin teksturoinnissa runsaasti varsinkin talojen ja pienempien ympäristön asettien teksturoinnissa. Vanhemman teksturointitavan mukaisesti okklusiokartta lisättiin diffuusiokarttaan helpottamaan teksturointia ja jotta lopputuloksesta saataisiin valaistukseltaan näennäisesti realistisempi. Diffuusiokartan päälle lisätty okklusiokartta lisäsi objektien istuvuutta ympäristöön, vähentäen samalla varsinkin suurten pintojen ilmeettömyyttä. Okklusiokartan käyttäminen nopeutti työskentelyä, kun kaikkia varjostuksia ei tarvinnut maalata tekstuuriin itse.

5.10 Emissiivisyyskartta

Emissiivisyyskartalla voidaan luoda illuusio valoa hohtavasta materiaalista ilman oikeaa valonlähdettä. Emissiivisyyskartat ovat halpa tapa luoda vaikutelma valoa erittävästä materiaalista, vaikka tosiasiaassa materiaalista ei tule valoa. Emissiivisyyskarttaa käyttävät materiaalit ovat itsevalaisevia, eivätkä ne vastaanota yhtään valaistusta. (Epic Games 2017.) Kuva 18 esittää mustavalkoista emissiivisyyskarttaa sekä pallon 3D-mallia, jonka päälle emissiivisyyskartta on levitetty. Emissiivisyyskartassa olevat valkoiset alueet voidaan nähdä oranssina hohtona pallon pinnalla.



KUVA 18. Pallo ja emissiivisyyskartta (Goo Create Learn 2016)

Emissiivisyyskartassa valoa hohtamattomat alueet ilmoitetaan mustalla ja hohtavat alueet valkoisen eri asteilla. Mitä valkoisempi alue on, sitä kirkkaampana alue hohtaa. Hohtoa on mahdollista myös sävyttää käyttämällä eri värisävyjä hohtavissa alueissa.

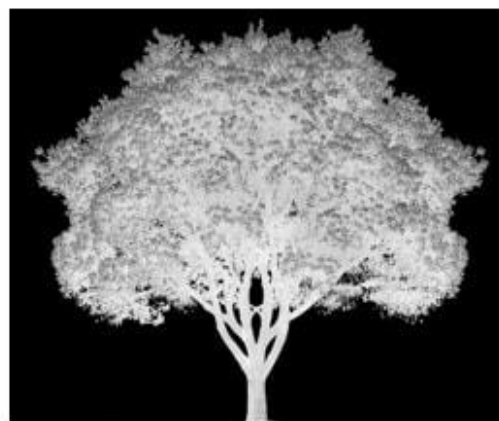
(Polycount 2017.) Emissiivisyyskartan luoma hohto ei kuitenkaan valaise dynaamisia objekteja, kuten pelihahmoja samalla tavalla kuin varsinaiset valot. (Epic Games 2017.)

5.11 Läpikuultavuuskartta

Läpikuultavuus- tai opasiteettikarttaa (engl. opacity map) käytetään kertomaan pelimoottorille, kuinka läpinäkyviä tekstuurikartoissa olevat pikselit ovat. Läpikuultavuuskartalla on mahdollista tehdä pikseleistä läpinäkymättömiä, läpinäkyviä tai jotakin siltä väliltä. Yleisesti ottaen läpikuultavuuskarttaa käytetään, kun joistakin alueista objektissa halutaan tehdä läpinäkyviä. Tällaisia käyttötarkoituksia voivat olla ikkunat tai efektit, joissa tarvitaan eri asteita läpikuultavuutta, kuten tuli, savu tai vaikkapa seinissä olevat luodinreiät. Läpikuultavuutta käytetään myös peleissä yleisesti partikkeliefekteissä, kuten sateessa tai kipinöissä. (Ahearn 2008, 83, 86.) Kuva 19 esittää puun bittikarttakuvaa sekä sen mustavalkoista läpikuultavuuskarttaa. Puuta ja sen lehvästöä ympäröivä alue saadaan läpinäkyväksi piirtämällä häivytettävät osat mustalla.



Tree



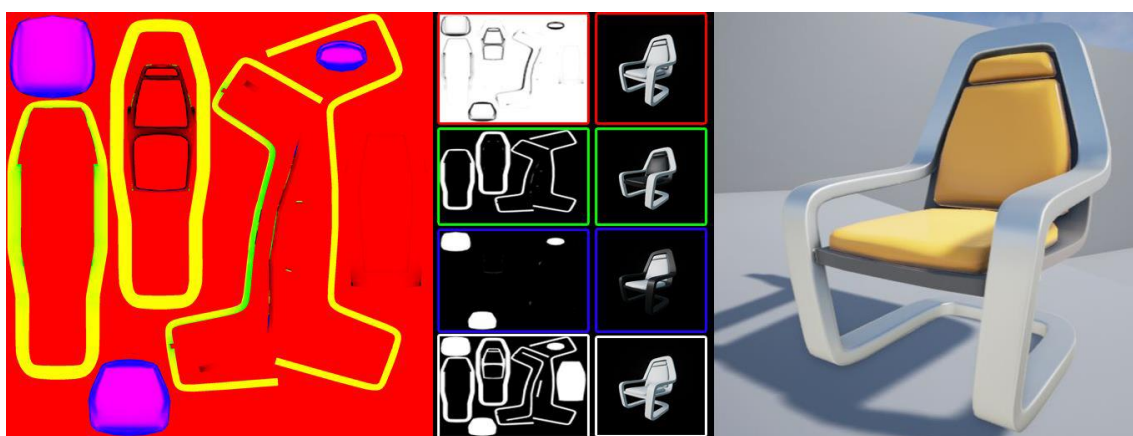
Tree Opacity Map

KUVA 19. Puu ja sen läpikuultavuuskartta (Pandey 2016)

Läpikuultavuuskartta on harmaasävyinen ja se käyttää yleensä erillistä kanavaa, kuten 32-bittisen bittikartan alfakanavaa määritelläkseen pikselien läpikuultavuuden. Läpikuultavuuskarttaa on myös mahdollista käyttää erillisenä kuvatiedostona varjostimessa. Alfakanavia käyttävät bittikartat ovat tiedostokooltaan kuitenkin yleisesti isompia. (Ahearn 2008, 83, 86.)

5.12 Peittomaskit

Peliobjekteja luotaessa voi olla joskus tarpeellista pystyä määrittelemään eri pintojen ominaisuuksia saman materiaalin sisällä. Helppo tapa tämän saavuttamiseksi on käyttää peittomaskeja. Peittomaskit (engl. texture mask) määrittelevät, mihin alueisiin materiaalin eri osat vaikuttavat objektin pinnalla. Peittomaskit ovat erittäin tehokas tapa luoda runsaasti vaihtelua peliobjekteissa, käyttämällä silti hyvin pienen määrän työtä tekstuuriden luomiseen. (Epic Games 2017.) Kuvassa 20 on tuolin 3D-malli sekä siinä käytetty peittomaski. Tuolin eri osat saadaan renderöityä eri materiaaleiksi värikoodaamalla ne peittomaskiin.



Kuva 20. Tuolin peittomaski ja renderöity kuva tuolista materiaalien kanssa (Epic Games 2017)

Peittomaski on harmaasävyinen tai yhtä värikanavaa käyttävä tekstuurikartta, jota käytetään rajoittamaan materiaalin käyttäytymistä peliobjektin pinnalla. Usein peittomaski sisällytetään jonkin tekstuurin läpinäkyvyys- tai värikanavalle, kuten diffuusio- tai normaalikartan alfa-kanavalle. Peittomaskeja voidaan luoda missä tahansa kaksiulotteisessa kuvankäsittelyohjelmassa. Alueet, joihin materiaalin vaikutus halutaan kohdistuvan, maalataan esimerkiksi valkoisella värillä ja alueet, joiden halutaan pysyä muuttumattomana, maalataan toisella värillä, esimerkiksi mustalla. (Epic Games 2017.)

Rescue 2 -projektin kehityksessä peittomaskeja käytettiin muun muassa luomaan vaihtelevuutta ympäristön objekteihin ilman, että artistien olisi tarvinnut luoda monta erilaista tekstuuria tai käydä muokkaamassa jokaista peliobjektia erikseen. Autojen rungon värittämisessä käytettiin mustavalkoista peittomaskia, johon värikoodattiin ne alueet, joiden

väriin haluttiin vaikuttaa. Tällöin ainoastaan valkoisella värillä maalattujen alueiden väri muuttui ajoneuvon pinnalla. Peittomaskin avulla artistien ei tarvinnut siis luoda monta erilaista tekstuuria jokaista eriväristä ajoneuvoa varten. Satama-alueen rahtikonteissa peittomaskia käytettiin määrittelemään, millaisia materiaaleja kontin pinnoilla esiintyy. Eri tyyppiset materiaalit värikoodattiin punaisella ja vihreällä värillä, jolloin varjostin osasi renderöidä nämä eri tavoilla peliä ajettaessa.

5.13 Tekstuuriatlakset

Kun pelimaailmaa rakennetaan, tekstuuriantisti tulee luoneeksi monta tekstuurikarttaa monelle kolmiulotteiselle objektille pelimaailmassa. Kun tämä kyseinen pelimaailma sitten ladataan ja ajetaan pelimoottorissa, pelimoottorin on haettava, eli niin sanotusti kutsuttava jokaista tekstuuritiedostoa jokaista kuvaa varten, minkä pelimoottori piirtää ruudulle. Jokainen näistä kutsuista hidastaa pelin ajoa, joten suorituskyvyn parantamiseksi on toivottavaa, että kutsujen määrää saadaan laskettua. (Ahearn 2008, 8-9.)

Kutsujen määrää voidaan vähentää esimerkiksi hyösyntämällä tekstuuripakkausta eli tekstuuriatlasointia. Pohjimmiltaan tämä tekniikka toimii niin, että iso määrä tekstuurikarttoja, jotka liittyvät jotenkin toisiinsa (yleisesti samassa ympäristössä olevat objektit) laitetaan yhteen isoon tekstuuritiedostoon. Tällöin, kun pelimoottori tarvitsee jonkin objektin tekstuurikarttaa piirtääkseen sen, sen on kutsuttava vain yhtä tekstuuria, josta se sitten hakee oikean alueen. (Ahearn 2008, 8-9.) Kuva 21 esittää kasveista koostuvaa tekstuuriatlasta.



KUVA 21. Tekstuuriatlas (Ahearn 2009, 83)

Tekstuuriatlaksia voi luoda joko käsin tai erillisillä työkaluilla. Työkalujen pääasiallinen etu on se, että niillä atlaksien luominen ja korjailu on nopeaa. (Ahearn 2008, 8-9.) Käsin tekemällä tekstuuriartistin on kasattava bittikarttakuvat oikeaan järjestykseen atlaksen sisään ja tämän jälkeen aseteltava objektien UV-saarekkeet bittikarttakuvaa vastaaville paikoille.

Tekstuuriatlaksia hyödynnettiin Rescue 2-projektissa monesti tilanteissa, joissa saman pelialueen maastossa käytettiin useampia pienikokoisia peliassetteja. Esimerkiksi satama-alueilla käytettäviä pienempiä peliobjekteja teksturoitiin samaan tekstuuriatlakseen, jolloin usean objektin tekstuuri oli mahdollista löytää nopeasti yhdestä tiedostosta ja objektien piirtämiseen tarvittavien kutsujen määrä saatiin minimoitua.

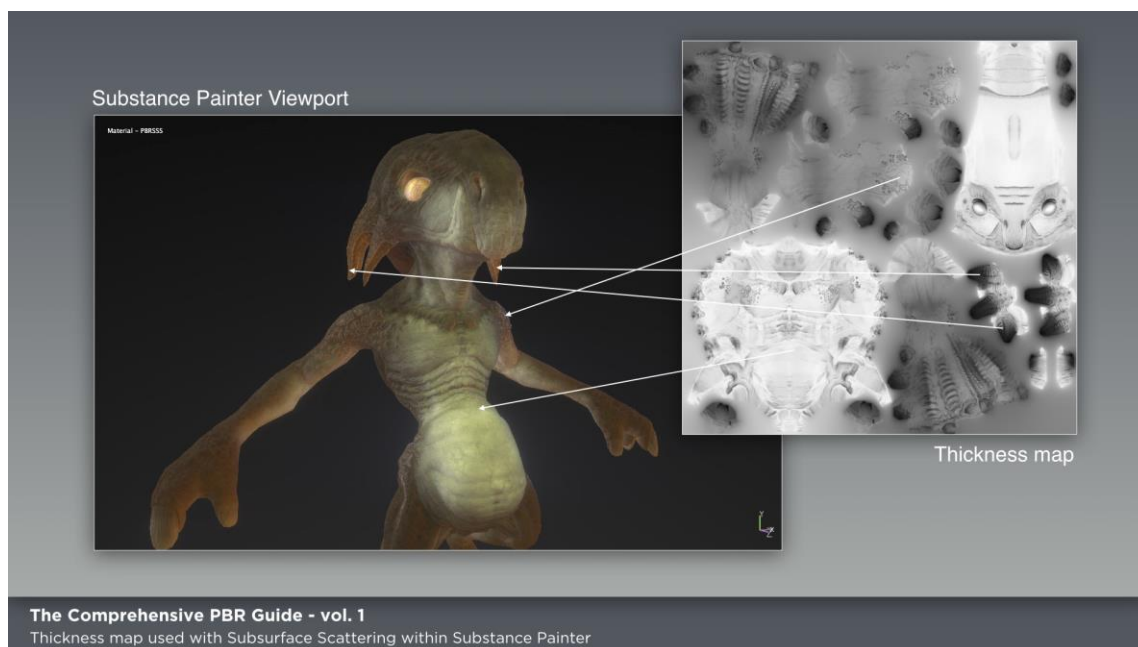
Pienten asettien teksturoiminen omiin karttoihinsa olisi ollut epäkäytännöllistä, useamman kartan lisätessä kutsujen määrää. Pienempien peliobjektien UV-saarekkeet harvoin täyttivät koko tekstuurikartan alueen, joten tästäkin syystä useampia asetteja oli järkevää mahdollistaa samaan tekstuuriatlakseen hukkatilan välttämiseksi. Useassa ta-

pauksessa atlaksien käyttö nopeutti työskentelyä, sillä useampia malleja voitiin teksturoida yhteen karttaan dokumenttia vaihtamatta. Tekstuurien nimeäminen ja löytäminen projektista oli nopeampaa, kun jokaista tekstuuria ei tarvinnut nimetä erikseen.

5.14 Tiheyskartta

Pinnanalainen sironta (engl. sub surface scattering) on termi, jota käytetään valoilmios-
tä, jossa valo siroaa sen läpäistessä läpikuultavan tai puoliksi läpikuultavan pinnan.
(Epic Games 2017.) Kun saapuva valo menee objektin pinnan sisään, se siroaa sen sisäl-
lä ja poistuu pinnalta eri paikassa. Tämän valoilmion simulointi on tärkeä osa realistisen
ihon luomisessa ja läpikuultavuuden simuloimisessa materiaaleissa, kuten vaha ja mar-
mori. (RenderMan University 2017.)

Kuva 22 havainnollistaa tiheyskartan vaikutusta 3D-mallin pinnassa valon osuessa sii-
hen. Ohuempien kehonosien voidaan nähdä olevan läpikuultavampia, kuin paksummat
kohdat.



KUVA 22. 3D-malli, joka käyttää tiheyskarttaa (McDermott 2017, 3)

Pinnanalaisen sironnan simuloimisessa käytetään tiheyskarttaa (engl. thickness map).
Tiheyskartta luodaan melkein samalla tavalla kuin ympäristön okluusiokartta, mutta

tiheyskarttaa leivottaessa pinnan normaalit osoittavat vastakkaiseen suuntaan. Tällöin bittikarttaan piirtyneet tummat alueet kuvastavat 3D-mallin ohuita osia ja valkoiset alueet 3D-mallin paksuja kohtia. (Allegorithmic Documentation 2017.)

Lopputuloksena saatua bittikarttaa käytetään varjostimessa pelimoottorin sisällä, jolloin varjostin osaa laskea, kuinka pitkälle saapuva valo matkaa objektin pintaan osuessaan. Vaihtoehtoisesti tiheyskartta on myös mahdollista sekoittaa diffuusiokarttaan pinnan-alaisen sironnan jäljittelemiseksi. Tällä tavalla saatu efekti ei tosin käyttäydy saapuvan valon mukaisesti. (Allegorithmic Documentation 2017.)

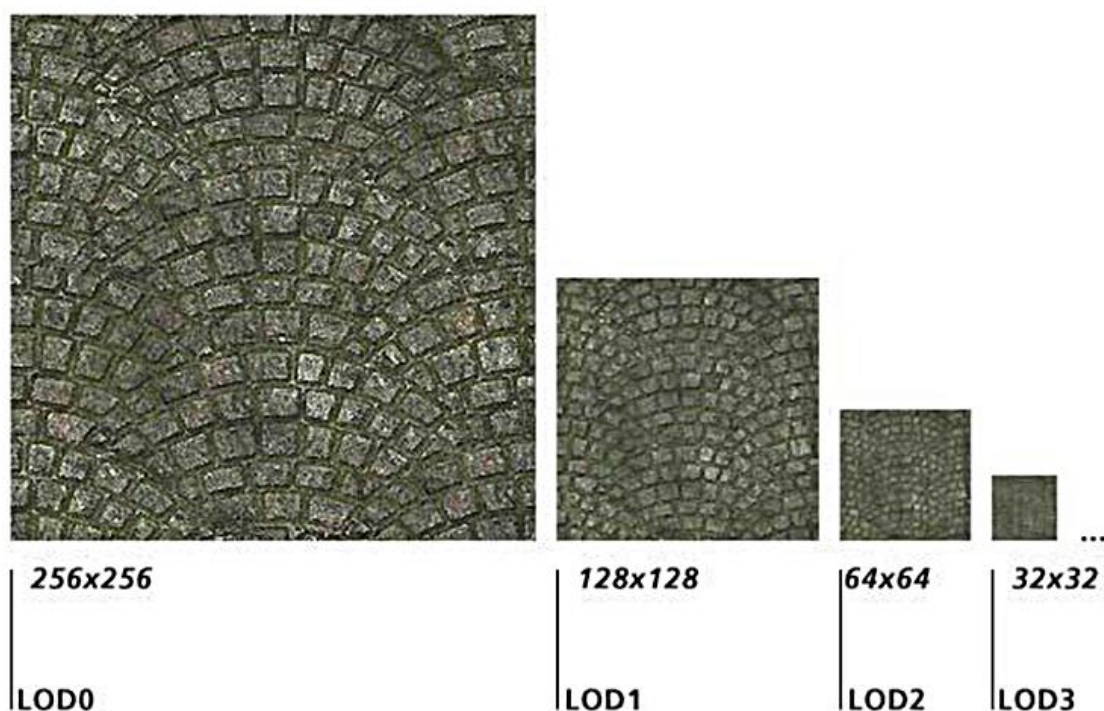
5.15 MIP-kartoitus

Koska teksturoituja objekteja voidaan tarkastella eri etäisyyksiltä pelissä, ongelmana on mitä resoluutiota tekstuurikartan tulisi käyttää. Liian korkea resoluutio kaukaa tarkasteltaessa johtaa pehmennysvirheisiin ja liian pieni resoluutio läheltä tarkasteltaessa jättää pois yksityiskohtia. Ratkaisuna on käyttää eri resoluutioisia tekstuurikarttoja, sen mukaan, mikä etäisyys objektin ja tarkastelijan välissä on. (Computer Graphics Library 2017.)

MIP -kartoitus (engl. MIP mapping), joskus myös tekstuurin LOD (level of detail, suom. yksityiskohtien taso), on yleensä ohjelmoijan hallitsema funktio, mutta joskus myös artistillakin voi olla mahdollisuus vaikuttaa tähän. MIP-kartoituksen käyttötarkoitusta voidaan havainnollistaa yksinkertaisella esimerkillä. Isoa ja pientä tekstuurikarttaa kaukaa tarkasteltaessa molemmat näyttävät samalta, sillä molempien piirtämiseen ruudulle käytetään sama määrä pikseleitä. Tämän takia ison tekstuurikartan käyttäminen objektissa, joka on kaukana kamerasta tuhlaa resursseja. Ison tekstuurin käyttäminen kaukana olevassa objektissa ei yleensä näytä yhtä hyvältä, kuin pienempi tekstuurikartta. Tämä johtuu siitä, että pelimoottori muuttaa tekstuurikartan kokoa ohjelman aikana, jolloin lopputulos on heikkolaatuisempi ja saattaa johtaa erilaisiin häiriöihin tekstuurin piirtämisessä. Näiden ongelmien poistamiseksi ohjelmoijat yleensä käyttävät MIP-kartoitusta. Termi MIP tulee latinankielisestä lauseesta ”multum in parvo”, joka tarkoittaa ”monia asioita pienessä tilassa”. (Ahearn 2008, 5.)

MIP-kartoituksessa alkuperäinen korkearesoluutioinen tekstuuri on skaalattu ja suodattettu eri kokoiisiin resoluutioihin. MIP-kartat voidaan automaattisesti generoida korkearesoluutioisesta tekstuurista, mutta tekstuuriartisti voi myös yksilöllisesti piirtää ne tai korjailla niitä. Tyypillisesti jokainen peräkkäinen MIP-taso on puolet edellisen koosta, joka takaa, että kokonainen tekstuuri (alkuperäinen sekä MIP-kartat) on korkeintaan puolitoista kertaa isompi kuin alkuperäinen tekstuuri. Jokainen skaalattu tekstuuri, tai MIP-taso esittää, miltä tekstuuri näyttää eri etäisyydeltä katsojan tarkastelupaikasta. Suodattimien avulla MIP-tasot antavat luonnollisemman ulkonäön, miten värit ja yksityiskohdat sekoittuvat toisiinsa, kun niitä tarkastellaan kaukaa. (VALVE Developer Community, 2015.)

Kuvassa 23 nähdään tekstuurikartan neljä LOD-tasoa. Vasemmalla oleva täysiresoluutioinen tekstuuri on 256 x 256 pikselin kokoinen ja jokaisella tasolla tekstuurin koko puolittuu, jolloin oikeassa laidassa olevan korkeimman LOD-tason koko on 32 x 32 pikseliä.



KUVA 23. MIP-kartoitettu tekstuuri ja sen neljä LOD-tasoa (Perrin 2014)

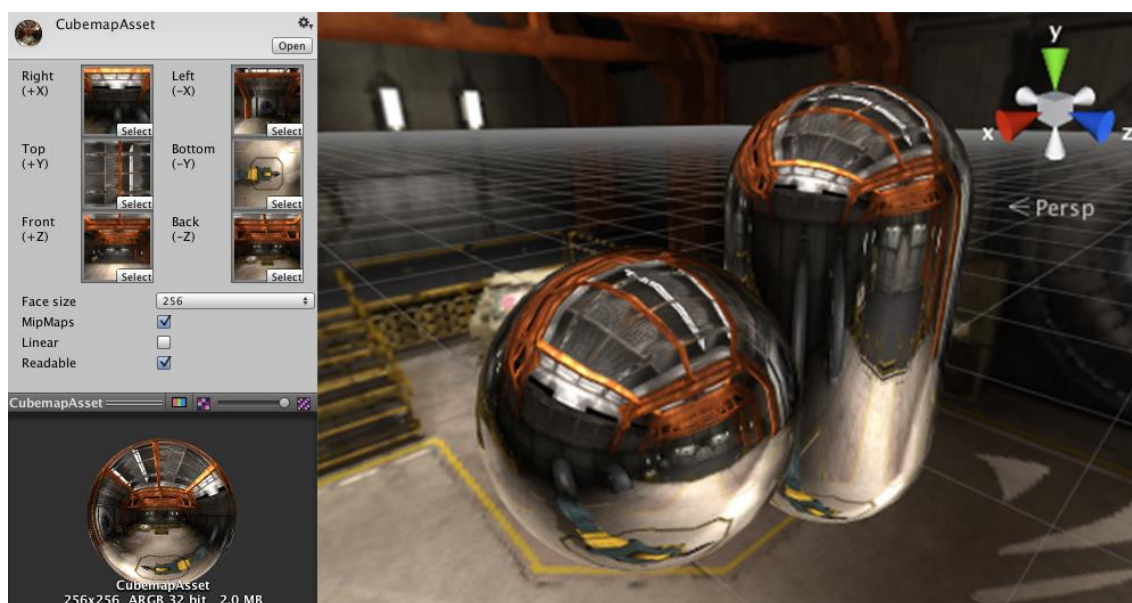
MIP-kartoituksen avulla voidaan säilyttää tekstuurin luettavuus kaukana kamerasta olevissa pinnoissa ja välttyä ns. moaree-interferenssikuvioinnilta, jota yleensä ilmenee pinnoilla, jotka ovat samassa kulmassa kameran kanssa. Koska MIP-kartat luodaan etukä-

teen, tämä tekniikka luo pienen korotuksen tekstuurien muistinkäytössä saavuttaakseen huomattavan korotuksen lopputuloksen laadussa. Tällä tekniikalla myös prosessoria kuluttavilta ajonaikaisilta laskuilta säästytään. (VALVE Developer Community, 2015.)

5.16 Ympäristökartta

Pelimoottorissa on monta erilaista tapaa luoda heijastuksia, mutta yleisin ja kaikista helpoin tapa toteuttaa niitä, on käyttää ympäristökarttaa. Ympäristökartta (engl. cubemap) on sarja kuvia, joita käytetään jäljittelemään ympäristön heijastuksia, jotka näkyvät objektien pinnoilla. (Ahearn, L. 2008, 5.)

Kuva 24 esittää ympäristökarttaa Unity-pelimoottorissa. Vasemmalla nähdään ruutu, jossa ympäristökarttaan tulevat kuvat valitaan ja oikealla heijastusten lopputulos kahden kappaleen pinnalla.



KUVA 24. Unity-pelimoottorissa koostettu ympäristökartta (Unity Technologies 2017)

Ympäristökarttojen nimi tulee siitä, että objektien pinnoilla nähdyt heijastukset ovat kuusi kuvaa järjesteltyinä kuution tahkoille sen sisäpuolelle ja heijastettuina kiiltävän objektin pinnalle. Nämä kuvat on piirretty objekteja ympäröivästä tilasta, jolloin ympäristökartta heijastaa näiden ympärillä olevaa aluetta mahdollisimman tarkasti. Piirretyt kuvat kattavat kaikki suunnat. Parhaan lopputuloksen aikaansaamiseksi niiden pitäisi ryhmittyä jonoon, jolloin muodostuva heijastus olisi saumaton. Ympäristökartan kuvat

ovat yleensä staattisia, jolloin pinnoissa näkyvä heijastus on aina samanlainen. Tämä tarkoittaa sitä, ettei esimerkiksi heijastukseen katsova pelihahmo voi nähdä itseään heijastuksessa. Tämä tapa on kaikista tehokkain suorituskyvyn maksimoimiseksi, mutta on olemassa myös tapoja, joilla reaaliaikainen heijastusten luominen on mahdollista. (Ahearn, L. 2008, 5.)

Yksi näistä tekniikoista on niin sanottu dynaaminen ympäristökartoitus (engl. dynamic light mapping). Tämä tekniikka piirtää ympäristökartan kuvat uudelleen jokaisessa ruudussa. Jos ympäristökartassa oleva objekti liikkuu tai jokin sitä ympäröivässä tilassa liikkuu, ympäristökartta päivitetään, jotta heijastukset pystyvät tarkasti jäljittelemään ympäristöä reaaliajassa. (Ahearn, L. 2008, 5.)

Jotkin pelimoottorit tukevat myös pallonmuotoisten ympäristökarttojen käyttämistä. Pyöreässä ympäristökartassa (spherical environment map) ympäristön kuvat asetellaan pallon sisäpuolelle. (Unity Technologies, 2017.)

6 TEKSTUURIEN VISUAALISET TYYLIT

On perusteltua todeta, etteivät kaikki tietokonepelit näytä keskenään samalta. Pelin lajityyppi, aihe, sanoma ja projektiin valittu tyyli ovat kaikki tekijöitä, jotka vaikuttavat siihen, miltä lopullinen tuote tulee näyttämään. Pelissä käytetyllä artistisella tyylillä on mahdollista viestiä sen sisällöstä sekä herättää pelaajassa erilaisia tunteita. (Novak 2012, 170.)

Pelin lajityyppi asettaa osittain ennakko-odotuksia sen ulkoasulle, joten hyvin suunniteltu ja toteutettu visuaalinen tyyli auttaa antamaan pelin sisällölle ja sen sanomalle lisäuskottavuutta. Pelimaailma ja sen hahmot luodaan yleensä tyylillä, joka sopii pelin muuhun sisältöön ja teemaan. Yhdenmukainen tyyli auttaa ylläpitämään pelin tunnelmaa. Tyyleillä on mahdollista tehdä myös vastakkainasetteluita pelin teeman kanssa. (Novak 2012, 170.)

Teksturointitekniikoiden näkökulmasta kolmiulotteisissa peleissä käytetyt visuaaliset tyylit voidaan lajitella neljään eri kategoriaan. Nämä ovat fotorealistinen teksturointi, puolirealistinen teksturointi, käsinmaalattu teksturointi sekä erilaiset abstraktit teksturointityylit. (Coleman 2017.) Näiden lisäksi on olemassa myös algoritmeihin perustuva proseduraalinen teksturointi. Edellämainittuja tyylejä on mahdollista sekoittaa toisiinsa ja on olemassa useita pelejä, joiden visuaalinen tyyli on jonkinlainen välimalli edellä mainituista. Tämän takia kaikkia tyylejä ei ole mahdollista sovittaa tiukasti näihin lokeihin.

6.1 Realistinen teksturointi

Realistinen tyyli pyrkii jäljittelemään oikeaa maailmaa valokuvien tavoin. Tällaista tyyliä luotaessa on otettava huomioon tärkeitä seikkoja, kuten väripaletin käyttö sekä riittävä taso yksityiskohdille. Kaikkea oikeassa maailmassa nähtyjä mikroskooppisia yksityiskohtia ei ole mahdollista jäljentää, mutta realistinen tyyli esittää asiat sellaisella tavalla, joka jäljittelee sitä, kuinka asiat oikeassa elämässä näemme. Pelimaailmaa luotaessa on huomioitava asioita kuten kohoumat, heijastumiset ja valon siroaminen. Realistinen peliobjekti tarvitsee kaikkia näitä ominaisuuksia luodakseen aitouden tuntua lop-

putulokseen. (Demers 2002, 167.) Kuvassa 25 nähdään kuvankaappaus Electronic Artsin alaisuudessa toimivan DICE -studion kehittämästä Battlefield 1 -pelistä, joka pyrkii teksturoinnissaan erittäin realistiseen ulkonäköön. Grafiikoiden tuottamisessa käytettiin apuna fotogrammetriaa, joka takasi tekstuurien korkean laadun (Tovar 2016).



KUVA 25. Kuvankaappaus Battlefield 1 -pelistä (Flickr 2016)

Realistinen tyyli on kaikista vaativin ja haastavin visuaalinen tyyli, sillä se vaatii erittäin harjaantuneen artistin, jotta mittasuhteet, tekstuurit ja 3D-mallit saadaan kaikki oikeanlaisiksi. Se on myös hyvin aikaavievä tyyli objektien luomisen kannalta, koska 3D-mallit ja tekstuurit ovat yleensä huomattavasti yksityiskohtaisempia kuin muissa visuaalisissa tyyleissä. Tätä tyyliä käyttävät muun muassa Grand Theft Auto-, Battlefield- ja Call Of Duty -pelisarjat. Realistisella tyylillä pyritään yleensä saamaan aikaan elokuvamainen ja immersoiva pelikokemus. (Coleman 2017.)

Realistisessa tyylissä teksturointi on usein valokuviin perustuvaa (Paquette 2009, 86). Valokuviin perustuvassa teksturoinnissa valokuvissa olevat oikean maailman ympäristöt muutetaan tekstuureiksi peliin sopivaan muotoon (Schultz 2017). Valokuvia käytettäessä on kuitenkin muistettava, että ne tallentavat vain lopputuleman siitä, miten valo käyttäytyy matkatessaan ympäristössä. Koska ne eivät mittaa objektien väriä, vaan heijastuvan valon väriä, joka perustuu lukuisiin eri muuttujiin, voivat valokuvissa nähdyt väritiedot olla virheellisiä. Ensimmäinen vaihe valokuvasta saatavan tiedon ulos saamiseksi, onkin oppia erottamaan sen sisältämä tieto ja miten sen voi eritellä kuvasta niin, että sen voi muuttaa pelimoottorissa käytettävään muotoon. (Paquette 2009, 86.)

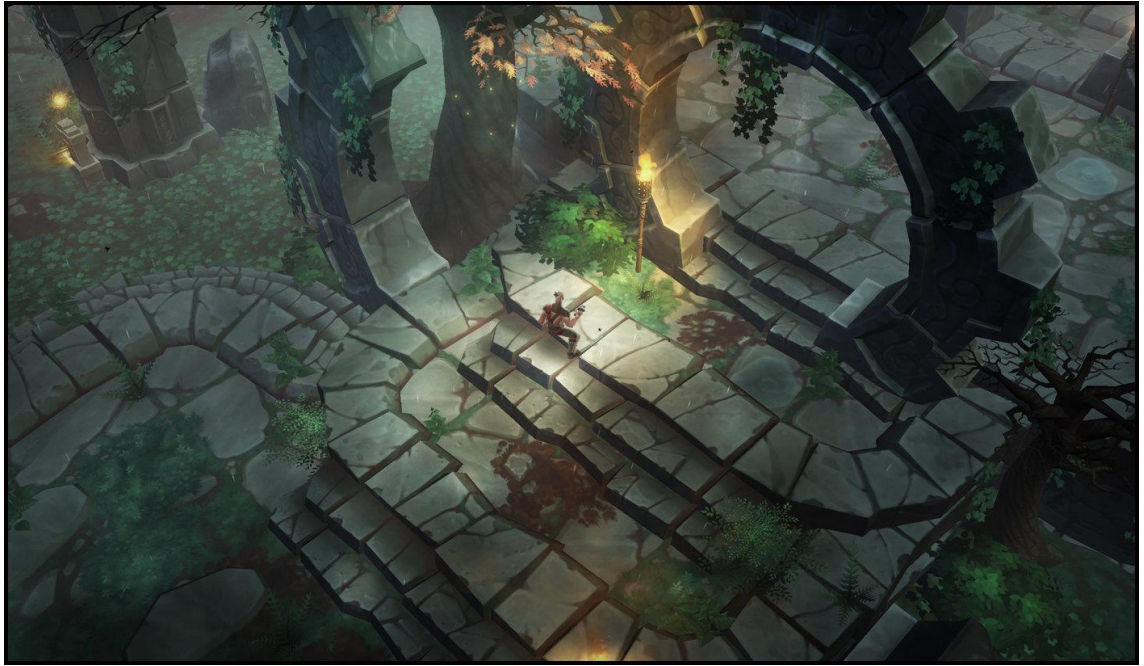
Jos kuvareferensseinä käytetään digikameroilla otettuja valokuvia, voidaan ne avata kuvankäsittelyohjelmassa, jossa niitä muokataan paremmin peliin sopiviksi. Valokuvien kontrastia ja värejä on mahdollista säätää, jolloin kuvista voidaan poistaa virheellistä väri- ja valoinformaatiota. Näiden lisäksi kuvista on mahdollista poistaa toistuvia elementtejä, joihin pelaajan katse kohdistuisi joko muuttamalla niitä huomaamattomiksi tai muokkaamalla niiden päälle elementtejä, jotka eivät erotu niin selkeästi. (Gantzler 2005, 40-45.)

Valokuvaan perustuvien tekstuurien lisäksi tekstuurikarttoja on myös mahdollista luoda alusta alkaen, ilman erinäistä kuvareferenssiä erilaisilla kuvankäsittelyohjelmilla (Schultz 2017).

6.2 Käsinmaalattu teksturointi

Käsinmaalatut, tyylitellyt tekstuurit edustavat tyyliä, jota pelistudiot ovat alkaneet hiljalleen omaksua uudelleen pelin visuaalista ilmettä valitessaan. Käsinmaalatussa tekstuurissa mallin päälle maalataan perinteisellä digitaalisella maalaustyyllillä. Lopputuloksena on pelimootorissa käytettävä perinteinen diffuusiokartta. (Bubenová 2016, 6-7.)

Käsinmaalatussa tyyliä muita karttoja, kuten normaali- tai spekularisuuskarttoja ei käytetä yhtä suuressa määrin kuin realistisessa tyyliä. Pelit, jotka käyttävät tätä tyyliä ovat yleensä tyyliteltyjä, fiktionaalisia pelejä, kuten World of Warcraft. Realistisempaa lähestymistapaa käyttävissä peleissä tällaista tyyliä ei juuri käytetä. Erona muihin teksturointitapoihin on yleinen artistinen tyyli ja ilme, joka valmiissa peliobjektissa on. Käsinmaalatut tekstuurit yhdistetään usein fantasiamaailmoihin, toisin kuin realistinen lähestymistapa, joka kuvastaa paremmin oikeaa maailmaa. (Bubenová 2016, 6-7.) Kuvassa 26 on kuvankaappaus Runic Gamesin tuottamasta Torchlight 2 -pelistä, joka hyödyntää käsinmaalattua tyyliä tekstuurigrafiikassaan.



KUVA 26. Kuvankaappaus Torchlight 2 -pelistä (ArtStation 2015)

Toinen iso ero käsinmaalatussa tyyliässä on käytettyjen tekstuurikarttojen määrä. Useimmat teksturointityylit käyttävät vähintään kolmea erilaista tekstuurikarttaa, kuten normaali-, siirtymä-, tai spekularisuuskarttaa. Koska käsinmaalatussa tyyliässä valo- ja varjoinformaatio on sisällytetty yhteen diffuusiokartassa maalamalla nämä käsin tekstuuriin, usealle erilaiselle tekstuurille ei ole välttämättä tarvetta. Joskus ympäristön okluusiokarttaa käytetään lähtökohtana maalausprosessissa. (Bubenová 2016, 7-8.)

Tapa, jolla tekstuurit on maalattu, on vaihtunut huomattavasti. Ennen käsinmaalatut tekstuurit näyttivät likaisemmilta ja levottomimmilta, jolloin pelaajalla oli vaikeuksia löytää objektin huomiopiste. Yksinkertaisemman lähestymistavan ansiosta tekstuurit näyttävät nykyään paremmilta, kun objektin pinnalla on vähemmän sekavuutta aiheuttavia yksityiskohtia. (Bubenová 2016, 7-8.)

Käsinmaalatuilla tekstuureilla on tavallisesti kolme erilaista erityispiirrettä. Muodoissa olevat reunaviivat ovat yleensä hyvin täsmennettyjä. Tällä tekniikalla erilaiset muodot on helppo nostaa esiin tekstuurista. Väriliu'ulla tai gradientilla kutsutaan siirtymää yhdestä väriarvosta toiseen. Tämä tekniikka on yksinkertainen, mutta se auttaa kiinnittämään katsojan silmän sinne, minne teksturiartisti sen haluaa. (Fletcher 2012, 47.)

Gradientit valaisevat alueita ja tuovat lämpöä sekä vaihtelevuutta tekstuuriin. Maalamalla viistoja reunoja pinnoille, tekstuuriin saadaan tuotua kolmiulotteista vaikutelmaa, jolla saadaan lisättyä paksuutta ja massaa objektin pintaan. Tällä tavalla voidaan myös luoda pieniä yksityiskohtia pinnoille. (Fletcher 2012, 47.)

Käsinmaalattuja tekstuureita luodaan kuvankäsittelyohjelmissa, kuten Photoshopissa samalla tavalla kuten digitaalisia maalauksia, käyttämällä pehmeitä ja kovia siveltimiä. Piirtopöytä on yleisesti lähes välttämätön väline prosessissa, sillä se osaa simuloida siveltimen painetta. Kuvankäsittelyohjelmien lisäksi tekstuureita voidaan myös maalata suoraan objektin pintaan kolmiulotteisessa näkymässä sellaisissa ohjelmissa, kuten ZBrush ja BodyPaint 3D.

6.3 Puolirealistinen teksturointi

Puolirealistinen teksturointi eli tyylielty realismi sekoittaa realistista teksturointityyliä sekä käsinmaalattuja tekstuureita. Puolirealistinen teksturointi lainaa elementtejä realistisesta tyylistä ja lisää siihen tyylieltyä ilmettä. Raja realistisen ja tyylieltyyn teksturoinnin välillä ei ole selkeä ja eri peleissä voidaankin nähdä eri asteita realismia sekoitettuna muuhun visuaaliseen ilmeeseen. (Anhut 2016.)

Puolirealistinen teksturointi perustuu osittain realistiseen teksturointiin, sekoittaen sekä käsinmaalattuja- että realistisia teksturointityylejä. Tyyliä on muokattu esittääkseen todellisuutta muistuttavaa maailmaa, mutta eroaa siitä silti enemmän tai vähemmän. Tieteisfiktioon tai sarjakuviin perustuvat pelit yleensä käyttävät tällaista teksturointityyliä. (Coleman 2017.)

Kuvassa 27 on tekstuurikartta sekä kuvankaappaus Arkane Studiosin kehittämästä Dishonored-pelistä, joka yhdistelee teksturointityylissään valokuvaan perustuvaa sekä käsinmaalattua teksturointia.



KUVA 27. Tekstuurikartta ja kuvankaappaus Dishonored-pelistä (ArtStation 2016)

Puolirealistisessa tyyliässä voidaan käyttää valokuviin perustuvia tekstuureita, joiden värejä ja kontrastia sitten säädetään kuvanmuokkausohjelmassa. Yleistä on, että kuvien päälle maalataan alkuperäistä valokuvareferenssiä jäljitellen, luoden näin sekoituksen molempien visuaalista ilmettä. Maalauksellinen tuntuma ja pehmeät korkeakontrastiset varjostukset vaikuttavat puolirealistisen tyylin yleisilmeeseen, tehden ympäristöstä ilmeikkään ja välttämättä monotonisuutta, jota voi joskus nähdä ympäristöissä ja objekteissa, joiden tekstuurit ovat puhtaasti valokuviin perustuvaa materiaalia. Alemmilla grafiikka-asetuksillakin hyvin toteutettu puolirealistinen tyyli voi olla näyttävä, vaikka pelistä riisuttaisiinkin suurin osa jälkikäsitteilyefekteistä. Tämä johtuu siitä, ettei peli luota pelkästään näihin grafiikkaehosteisiin. (Hawkins 2013.)

Yksi etu tässä tekniikassa on, että tekstuureissa on jonkin verran varjostusta itsessään, jolloin korkealle asetetut valaistus- ja varjostusasetukset parantavat ilmettä entisestään. Tämä varjostus myös korostaa ja parantaa yksityiskohtia, joita objekteihin on mallinnettu. (Hawkins 2013.)

Toinen etu kirkkaissa kontrasteissa ja pehmeissä väriliu'issa on, että teksturointi toimii hyvin myös alemmilla resoluutioilla. Vaikka tekstuurit eivät näytäkään yhtä hyvältä läheltä tarkasteltuna, kun terävyys katoaa, ne toimivat kaukaa hyvin. Vaikka tietokonegrafiikka kehittyykin jatkuvasti, hyvin toteutettuna, nämä pelit voivat säilyttää näyttävän ulkoasunsa, niiden vahvan visuaalisen tyylin ansiosta. (Hawkins 2013.)

6.4 Abstraktit teksturointityylit

Abstraktit teksturointityylit ovat omaleimaisempia pelien visuaalisista tyyleistä, sillä ne eivät muistuta muita tyylejä erityisen paljoa. Abstraktit tyylit voivat käyttää esimerkiksi yksinkertaisia geometrisia muotoja, jolloin abstraktien tekstuurien luominen ei yleensä ole niin aikaavievää, kuin muissa tyyleissä. (Coleman 2017.)

Monet abstraktia tyyliä käyttävät pelit hyödyntävät eläviä ja vahvoja värejä tai vaihtoehtoisesti yksinkertaistettua väripalettia pelin sisältöä vastaavan tunnelman ja tyylin aikaansaamiseksi. (Coleman 2017.) Kuvassa 28 on kuvankaappaus SUPERHOT Teamin kehittämästä SUPERHOT-pelistä, joka hyödyntää hyvin rajattua värimaailmaa ja yksinkertaistettuja muotoja visuaalisessa tyylissään.



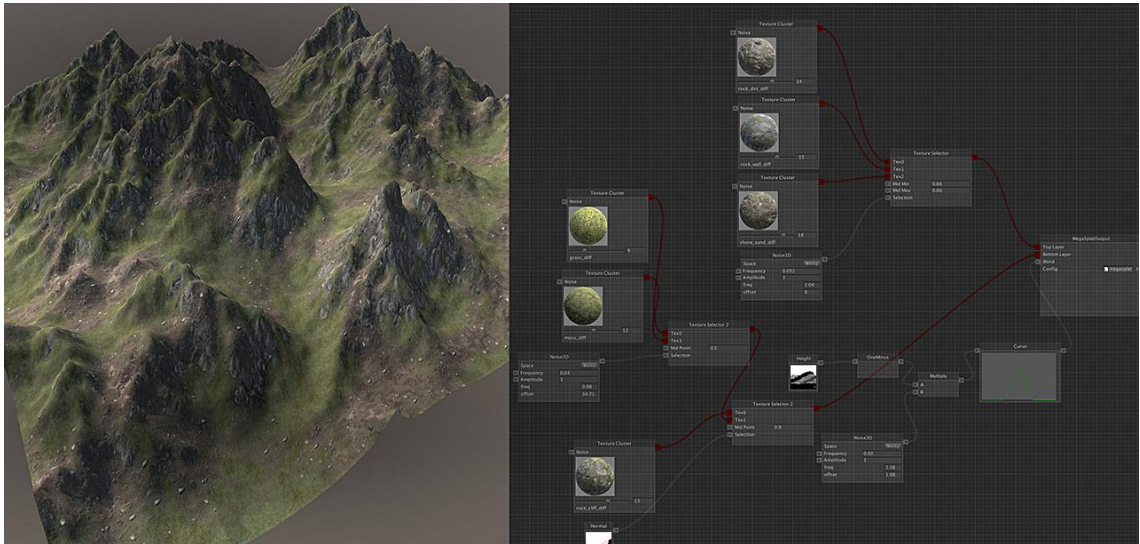
KUVA 28. Kuvankaappaus SUPERHOT-pelistä (Steam 2017)

6.5 Proseduraaliset tekstuurit

Proseduraaliset tekstuurit ovat algoritmeilla määriteltävä tapa esittää tekstuureita. Toisin kuin bittikarttoja käyttävät tekstuurit, proseduraalinen tekstuuri esittää tekstuurin matemaattisesti luoduilla kuvioilla. Proseduraaliset tekstuurit ovat riippumattomia resoluuti-

osta, jonka takia ne voivat luoda tarkempia tekstureita. Ne ovat hyödyllisiä varsinkin silloin, jos teksturoitavana on isoja alueita. (PC Magazine Encyclopedia 2017)

Kuva 29 esittää kuvankaappausta Unity-pelimoottorin MegaSplat-liitännäisestä. Vasemmalla oleva maasto on teksturoitu käyttämällä proseduraalista teksturointia, säätämällä oikealla näkyvässä editorissa olevia tietosolmuja.



KUVA 29. Unity-pelimoottorin MegaSplat-liitännäinen (MegaSplat 2017)

Yleisesti ottaen proseduraalisia tekstureita on suhteellisen helppo käyttää ja muokata, koska ne eivät tarvitse erityistä kartoitusta. Tästä huolimatta proseduraaliset tekstuurit voivat olla myös hyvin monimutkaisia. Proseduraaliset tekstuurit toimivat, kuten laatoituvat tekstuurit. Ne istuvat täydellisesti reunoista toisiinsa ja näyttävät siltä, miltä niiden on tarkoitus näyttää jopa silloin, kun niitä leikataan. (Blender 2.79 Manual 2017)

Proseduraalisessa teksturoinnissa on myös mahdollista käyttää bittikarttakuvia, joiden käyttäytymistä muokataan tietosolmuilla. Tekstuurikarttojen näkyvyyttä voidaan esimerkiksi satunnaistaa ja niitä voidaan sekoittaa toisiinsa. (Davis 2013.)

6.6 Tekstuurien tyyli Rescue 2: Everyday Heroes -pelissä

Rescue 2 -pelissä tekstuurigrafiikan visuaalinen tyyli voidaan luokitella puolirealistiseksi. Grafiikat hyödyntävät valokuviiin perustuvaa teksturointia, mutta useita yksityiskoh-

tia on vahvistettu maalaamalla käsin valokuvatekstuurien päälle, jotta alueita saadaan terävöitettyä ja yksityiskohtia nostettua esille. Monia elementtejä tekstuureissa on myös piirretty kokonaan itse käyttämällä valokuvateksturointia referenssinä tyyllitellymmän ja näyttävämmän ulkoasun saavuttamiseksi.

Pelin graafisessa tyyliässä tehtiin tietoinen ratkaisu pyrkiä pelin aikaisempaa osaa hie-
man tyyllitellympään linjaan. Tyyliässä pyrittiin tuomaan pelin ulkonäköön jopa sarjaku-
vamaisiakin piirteitä. Grafiikoissa hyödynnettiin niin sanottua kohotettua realismia, joil-
la pelin grafiikoissa pyrittiin valintoihin, jotka palvelivat pelattavuutta. Pienoismaallien
kaltaisen tarkkuuden vastapainoksi tekstuureissa ja 3D-malleissa pyrittiin korostamaan
värejä ja muotoja, jotta peliobjektien informatiivisuutta saataisiin korostettua. Tällaisilla
valinnoilla pelaajan huomio saatiin kiinnitettyä pelikokemuksen kannalta oleellisiin
asioihin ja kommunikoitua pelaajalle niiden käyttötarkoitusta paremmin.

Grafiikkaa tehtäessä liikuteltavien ja pelaajan kanssa vuorovaikutuksessa olevien ele-
menttien, kuten pelihahmojen ja ohjattavien ajoneuvojen tekstuureista pyrittiin teke-
mään helposti luettavia ja intuitiivisia. Tosimaailman esikuvia hyödyntävien objektien
mittasuhteissa ja ulkonäössä harrastettiin pelkistämistä ja tiivistämistä. Esimerkiksi pa-
loautojen pituutta lyhennettiin verrattuna niiden oikeisiin esikuviin ja niissä olevien va-
rustekaappien määrää vähennettiin. Palomiesten käyttämien varusteiden kokoa vastaa-
vasti liioiteltiin ja niiden osia värikoodattiin tekstuureissa, jotta pelaajan olisi helpompi
erottaa, miten päin kyseiset varusteet ovat ja mistä osista ne koostuvat.

Rescue 2:n edeltäjään verrattuna tekstuurien teknisessä toteutuksessa pyrittiin optimaai-
lisempiin ratkaisuihin. Varjostimia ja pelimootorissa tapahtuvaa asettien räätälöintiä,
kuten peittomaskien ja väripalettien käyttämistä peliobjektien jälkikäteen värittämisessä,
hyödynnettiin aikaisempaa osaa huomattavasti monipuolisemmin. Aikataulun salliessa
suoraan edeltäjäosasta uudelleenkäytettyjen grafiikka-asettien tekstuureita ja muotoa
käsiteltiin pelin visuaalisten periaatteiden mukaisesti ennen uudessa projektissa käyttä-
mistä, jotta ulkoasua saatiin yhtenäistettyä.

Tasapainottaessa tyyllitellyn ja fotorealistisen ulkoasun rajaa otettiin huomioon pelaaja-
kunnan realismihakuisuus. Tuotannossa pyrittiin säilyttämään tyylliteltä mutta tarpeeksi
realistinen linja. Koska Rescue 2 oli edeltäjänsä pelimäisempi, myös tekstuurien tyyliä
pyrittiin pitämään enemmän linjassa pelimekaniikan sisällön kanssa. Pelin graafisessa

ulkonäössä vertailtiin genren kilpailevaa Emergency-pelisarjaa, joka osaltaan myös ohjasi pyrkimyksiä tekstuurien tuottamisessa. Emergency-sarjan grafiikoiden laatuun pyrittiin, yrittäen samalla kuitenkin erottumaan siitä.

7 KAMERAKUVAKULMAN VALINTA

Tietokonepeleissä käytetään yleensä pelin lajityypille ominaista kuvakulmaa (engl. point of view, eli POV). Joissakin kuvakulmissa pelaaja näkee maailman pelihahmon silmin, toiset taas mahdollistavat pelihahmon tarkastelun kolmannesta persoonasta hahmon ulkopuolelta. (Novak 2012, 165.) Jotkin pelit voivat myös yhdistellä näitä kah- ta kuvakulmaa käyttämällä muuttuvaa kameraa, jolla kameran kulma ja etäisyys tarkas- teltavasta kohteesta on pelaajan säädettävissä. Tällöin kuvakulma voi vaihdella erilais- ten kolmannen persoonan kuvakulmien välillä tai vaihtua ensimmäiseen persoonaan ja takaisin. (Granberg 2014, 35.)

Jotkin pelit, kuten Grand Theft Auto IV sallivat jopa vaihtamisen pelaajaa seuraavan-, isometrisen-, ensimmäisen persoonan- ja ohjaamon kameran välillä. Muuttuva kamera voi olla joko pelaajan ohjaama tai vaihtoehtoisesti algoritmeilla ohjattu, tai näiden yh- distelmä. Pelaajaan ohjaamassa kamerassa pelaaja voi itse säätää näkymän kulman ja etäisyyden, suhteessa ruudulla näkyvään pelihahmoon. (Granberg 2014, 35.)

Algoritminen kamera perustuu taas ohjelmointiin, jolloin peli osaa itse säätää kuvakul- man ympäristön ja tapahtumien mukaan, kuten urheilupeleissä. Kamerakuvakulmien sekoituksessa pelaaja voi itse säätää kameran kulmaa ja etäisyyttä, mutta peli pystyy algoritmien avulla mukauttamaan kameraa, jotta pelaajan säätämä kuvakulma säilyy. (Granberg 2014, 35.)

Joissakin peleissä muuttuva kamera voi olla myös pelaajasta riippumaton, jos pelissä tapahtuu jotakin, joka vaatii kamerakuvakulman muutosta. Esimerkiksi Halo- pelisarjassa peliä tarkastellaan ensimmäisestä persoonasta, paitsi silloin, kun pelaaja astuu ajoneuvon. Tällöin kuvakulma vaihtuu kolmanteen persoonaan, jolloin pelaaja voi liikuttaa kameraa täydellä 360-asteen kehällä ajoneuvon ympärillä.

Pelin lajityypillä on yleensä iso vaikutus siihen, mitä kamerakuvakulmaa pelissä käyte- tään, sillä ei ole käytännöllistä käyttää peliin sopimatonta kuvakulmaa. Kamerakuva- kulma voi myös vaikuttaa siihen, miten tekstuureita tehdään ja millaisia tekstuurikartto- ja pelissä käytetään, sillä ensimmäisen ja kolmannen persoonan videopeleissä asioiden mallintamisella ja teksturoimisella on erilaiset vaatimukset (Chang 2006, IX).

7.1 Ensimmäinen persoona

Ensimmäisen persoonan kuvakulmassa (engl. first person POV) pelaaja näkee maailman pelihahmon silmin. Ensimmäisen persoonan ammutapelissä (engl. first person shooter, eli FPS) pelaaja näkee yleensä myös pelihahmon kädessä pitelemän aseensa ruudun alaosassa. Tämä kuvakulma voi joskus lisätä pelin immersiota pelaajan ja pelihahmon välille, sillä pelaaja astuu pelihahmon rooliin fyysisesti, eikä voi tarkastella tätä ulkopuolelta. (Novak 2012, 165.) Kuva 30 esittää kuvankaappausta suositusta Call of Duty: Black Ops 2 -pelistä, joka hyödyntää ensimmäisen persoonan kuvakulmaa.



KUVA 30. Kuvankaappaus Call of Duty: Black Ops 2 -pelistä (SG Gaming Info 2013)

Ensimmäisen persoonan kuvakulma on varsinkin nykypäivänä suosittu edellämäinitussa FPS -genressä. Muita lajityyppejä, jotka hyödyntävät tätä kuvakulmaa ovat yleensä erilaiset simulaatiot, ajopelit sekä seikkailupelit. Kuvakulman käyttäminen on varsin perusteltua peleissä, missä omaksutaan vain yhden hahmon rooli ja ohjaillaan tämän tekemisiä. Pelaajan ei tarvitse nähdä ympäristöä kuin vain sen verran, mitä luonnollisestikin henkilö pystyisi katseella havainnoimaan. Ensimmäinen persoona sopii hyvin peleihin, joissa pelikokemuksen halutaan olevan todentuntuinen ja toiminnantäyteinen ja pelaajan suorituksen tarkkuutta vaativa.

Ensimmäisen persoonan kuvakulmaa käyttävissä peleissä tekstuurikarttojen on oltava riittävän yksityiskohtaisia, mikäli peli pyrkii esittämään oikeaa maailmaa mahdollisimman realistisesti. Kuten oikeassa elämässäkin, pelaaja pystyy tarkastelemaan ympäristöä, sen objekteja sekä muita hahmoja erittäin läheltä. Tämä tarkoittaa sitä, että tekstuurikarttojen resoluution on oltava mahdollisimman korkea, jotta epätarkoilta ja suttuisilta ympäristöiltä vältytään. Realismia tavoittelevassa pelissä tällaiset seikat voivat helposti rikkoa pelaajan immersion ja kiinnittää tämän huomion epäoleellisiin.

Nykypäivänä ensimmäisen persoonan pelit käyttävät vähintään 2048 x 2048 pikselin kokoisia tekstuureja, jolloin riittävän tarkka yksityiskohtien taso on mahdollista saavuttaa ilman epätarkkaa jälkeä. 4096 x 4096 kokoiset tekstuurit ovat nykyään hyvin yleisiä tarkkuutta vaativissa objekteissa. Muun muassa tekstuurien ison koon takia ensimmäistä persoonaa käyttävät pelit vievät paljon tilaa ja vaativat tietokoneelta korkeampaa suoritustehoa.

Toisaalta ensimmäisen persoonan kuvakulman takia pelaajalla on usein kapeampi näkymä ympäröivästä maailmasta, jolloin kerrallaan piirrettävien tekstuurien määrä voi olla huomattavasti pienempi, kuin muita kuvakulmia käyttävissä peleissä. Tilanteissa, joissa pelaajan on mahdollista nähdä suurempia kokonaisuuksia kerralla, tietokoneen kuormitusta pyritään helpottamaan erilaisilla suorituskykyä nopeuttavilla ratkaisuilla, kuten esimerkiksi aikaisemmin mainitulla MIP-kartoituksella, piirtoetäisyyden rajaamisella tai jopa pienempien asettien piilottamisella pelaajalta.

7.2 Kolmas persoona

Kolmannen persoonan kuvakulmassa (engl. third person POV) pelaaja voi nähdä pelihahmon ruudulla. Tällä tavoin pelaaja voi säilyttää mielikuvan pelihahmon ulkonäöstä, mutta ei tarjoa täydellistä tunnetta pelihahmon ruumiin omaksumisesta ja maailman näkemisestä tämän silmin. (Novak 2012, 165.) Kuvassa 31 on kuvankaappaus Tom Clancy's The Division -pelistä, jossa pelaaja näkee tapahtumat kolmannesta persoonasta.



KUVA 31. Kuvankaappaus Tom Clancy's The Division -pelistä (Develop 2013)

Koska elokuvia harvemmin kuvataan ensimmäisen persoonan kuvakulmasta, on kolmannen persoonan peleissä hyvin elokuvamainen tuntuma. Voidaan väittää, että pelaaja muodostaa vahvemman tunnesiteen pelihahmoon, jonka hän voi nähdä ruudulla, kuin sellaiseen, jota nähdään vain harvoin tai jonka ainoa nähtävä ruumiinosa on asetta tai muuta objektia pitelevä käsi. Kun ruudulla näkyvää pelihahmoa luodaan, on erityisen tärkeää varmistaa, että pelihahmon ulkoasu (väriteema lukuun ottaen) on helposti erotettavissa muista ruudulla olevista hahmoista. Kolmannen persoonan seikkailupeleissä pelaajat voivat usein nähdä pelimaailman pelihahmon olkapään yli kuvattuna. (Novak 2012, 165.)

Kolmannen persoonan peleissä tekstuurien luomisessa pätee kutakuinkin samat periaatteet, kuin ensimmäisen persoonan peleissä. Koska kamerakuvakulma kuvaa tapahtumia lähes yhtä läheltä, kuin ensimmäisen persoonan peleissä, on tekstuurien tarkkuuksien oltava suuria, jotta riittävän laadukas lopputulos on mahdollista saavuttaa.

7.3 Muut kuvakulmat

Ensimmäisen ja kolmannen persoonan kuvakulman lisäksi peleissä käytetään myös erilaisia perspektiivejä ja kameroita. Sillä, millainen pelin lajityyppi on kyseessä, on suuressi vaikutusta käytettyyn perspektiiviin.

Kaikkialla läsnäolevassa tai niin sanotusti ubiikissa kuvakulmassa pelaajalla on kyky tarkastella eri alueita pelimaailmassa ja voi toimia monessa eri paikkaa maailmassa. Ubiikissa perspektiivissä pelaaja voi katsella maailmaan alaspäin ylhäältä käsin. Tällaisia pelejä ovat muun muassa reaaliaikastrategia- sekä ns. "jumalapelit". Ilmakuvakulmassa pelimaailma näytetään pelaajalle ylhäältäpäin kuvattuna lintuperspektiivissä. Tämä kuvakulma on suosittu alkuperäisissä Legend of Zelda ja Pac Man -peleissä. (Novak 2012, 223-224.)

Isometrisessä kuvakulmassa pelaaja pystyy näkemään hieman maiseman poikki noin 30-45 asteen kulmassa. Ennen reaaliaikaista kolmiulotteista piirtämistä, strategiapeleissä näkyvät objektit luotiin kaksiulotteisina bittikarttakuvina yläviistosta kuvattuna. Monet strategia- ja reaaliaikastrategiapelit luotiinkin isometrisestä kolmiulotteisesta kuvakulmasta ilman mahdollisuutta pyörittää kameraa. Ensimmäiset SimCity ja Civilization-pelit olivat melkein kokonaan ilmasta käsin kuvattuja, sillä sen aikaiset tietokoneet eivät pystyneet pyörittämään yksityiskohtaisempia grafiikoita. Myöhemmin nämä pelit omaksuivat isometrisen kuvakulman käyttämällä kaksiulotteista grafiikkaa, luoden näin valekolmiulotteisen vaikutelman, joka sopi paremmin pelien lajityypille. (Novak 2012, 224.) Kuva 32 esittää kuvankaappausta Blizzardin kehittämästä Diablo 3 -pelistä, jossa tapahtumia kuvataan ilmasta käsin.



KUVA 32. Kuvankaappaus Diablo 3 -pelistä (GamingCFG 2017)

Kaksiulotteisessa tilassa pelihahmo pystyy liikkumaan vain vasemmalle tai oikealle ja ylös- tai alaspäin. Pelihahmo ei pysty juoksemaan syvyyssuunnassa ruutua kohti tai pois päin siitä. Vanhat kaksiulotteiset pelit käyttivät sivuttaissuuntaista ruudun vieritystä luodakseen illuusion tilasta. (Novak 2012, 225.) Nykypeleissä voidaan hyödyntää tätä samaa kuvakulmaa, vaikkakin peliin on lisätty kolmas ulottuvuus. Sivukuvakulmassa tapahtumat nähdään sivultapäin kuvattuna vanhojen kaksiulotteisten tasohyppelyjen tapaan (Granberg 2014, 35).

Peleissä, joissa pelaaja joutuu ohjaamaan yhden pelihahmon sijasta useampia eri hahmoja tai vaihtoehtoisesti manageroimaan ympäristöä, on yleensä tavallista käyttää jotta-kin edellämämainituista ilmasta käsin kuvatuista kuvakulmista. Tällöin pelaajan on helpompi orientoitua tilanteeseen, jossa hänen on hallittava useampia hahmoja tai tilanteita samanaikaisesti, pitkilläkin välimatkoilla. Sanomattakin lienee selvää, ettei ympäristön tarkastelu katutasolta olisi esimerkiksi kaupunginrakennuspelissä kovin käytännönläheinen vaihtoehto, kun pelaaja joutuisi liikkumaan pitkiä matkoja rakennusten välissä yksinkertaistenkin toimintojen suorittamiseksi.

7.4 Kamerakuvakulma Rescue 2: Everyday Heroes -pelissä

Rescue 2-pelissä käytettiin ilmakuvakulmaa, jossa pelaaja pystyy tarkastelemaan pelialuetta ilmasta käsin viistossa kulmassa. Pelaajalla oli mahdollisuus säätää kamerakuvakulmaa loivemman ja jyrkemmän kulman välillä sekä siirtyä lähemmäksi tai kauemmaksi maatasosta. Tämän lisäksi pelaaja pystyi pyörittämään kameraa täydellä 360 asteen kehällä. Kuvakulman vaihtamisen avulla pelaaja pystyi paremmin havainnoimaan pelialuetta ja hallitsemaan pelihahmoja, olivatpa ne missä kohtaa aluetta tahansa. Koska pelin lajityyppi oli reaaliaikastrategia, jossa pelaaja joutuu liikuttelemaan useampia yksiköitä, oli ilmakuvakulman käyttäminen luonnollinen valinta. Kuvassa 33 on kuvankaappaus Rescue 2 -pelistä, jossa käytettiin ilmakuvakulmaa.



KUVA 33. Kuvankaappaus Rescue 2: Everyday Heroes -pelistä (Steam 2017)

Kameran kuvatessa tapahtumia ilmasta käsin, projektissa oli mahdollista käyttää resoluutioltaan hieman pienempiä tekstuurikarttoja, sillä pelaaja ei pystynyt tarkastelemaan maastoa liian läheltä, jolloin pieniresoluutioisten tekstuurien rakeus muuten tulisi esille. Illuusio realistiselta vaikuttavasta maailmasta rakentui ympäristön yleisilmeestä, sillä pelaaja havainnoi näkemäänsä kokonaisuutena. Tällöin jokaisen pelialueelta löytyvän objektin ei ollut tarpeellista olla tekstuuriltaan liiallisen tarkka vakuuttavan lopputuloksen aikaansaamiseksi.

Kaikille pelissä käytetyille objekteille ei ollut siis järkevää luoda samankokoista tekstuurikarttaa. Pienemmät ja kauempana kamerasta sijaitsevat objektit pystyivät käyttämään resoluutioltaan pienempiä tekstuurikarttoja, kuin isot ja lähellä kameraa olevat. Tämä säästi tekstuurien teossa käytettyä aikaa ja tietokoneen resursseja, sillä oli tarpeetonta käyttää aikaa yksityiskohtien teksturoimiseen, mitä pelaaja ei peliä pelatessaan pysty havaitsemaan. Tällä tavoin myös projektin tiedostokokoa ja pelin ajamista oli mahdollista keventää.

Tekstuureita luotaessa oli tärkeää ottaa huomioon niiden toimivuus pelin sisällä sekä samanlaisten kuvioden toistuminen pinnoilla, kun pelialueen kokonaisuutta tarkasteltiin. Tekstuurit, jotka toistuivat useasti esimerkiksi rakennusten pinnoilla, eivät saaneet sisältää liian huomattavia yksityiskohtia, sillä muuten niiden toistuminen erottuisi kau-

empaa tarkasteltaessa. Varsinkin maaston tekstuureissa liian näkyviä yksityiskohtia täytyi välttää ja tekstuureista täytyi tehdä eräänlainen keskiarvo, jotta toistuvat kuviot eivät osuisi pelaajan silmään. Erilaisten siirtokuvien (engl. decal) käyttämisellä oli mahdollista välttää pintojen monotonisuutta. Kameran etäisyys oli myös otettava huomioon yksityiskohtia teksturoitaessa. Liian tiheästi toistuvat kuviot aiheuttivat kohinaa 3D-mallien pinnoilla, kun niitä tarkasteltiin kauempaa. Tällöin esimerkiksi tiiliseinien tiileistä piti tehdä suurempia, kuin ne oikeasti ovat, jotta sotkuiselta ulkoasulta vältyttiin.

8 TEKSTUURIKARTTOJEN LUOMINEN ERI OHJELMILLA

Tekstuurikarttoja on mahdollista luoda lukuisilla erilaisilla ohjelmilla. Ohjelmat eroavat toisistaan niin ominaisuuksiltaan, kuin käyttöliittymiltään. Perinteisesti tekstuurikartat maalataan usein 3D-mallin geometriaa kuvaavien UV-piirrustusten avulla. Tekstuuritartistit työskentelevät kerroksittaisissa työtiedostoissa, joissa tekstuurin muokkaaminen on joustavaa. (Slick 2017.)

Perinteisten kaksiulotteisten kuvankäsittelyohjelmien rinnalle on kehitetty myös täysin teksturointiin ja tekstuurikarttojen generoimiseen keskittyviä ohjelmistoja. Jotkin ohjelmat, kuten Quixelin 3DO ja Allegorithmicin Substance Painter, mahdollistavat maalaamisen suoraan 3D-mallin pinnalle.

Käytetystä tekniikasta ja ohjelmasta riippuen tekstuurikarttoja voidaan tuottaa yksi kerrallaan tai useampia samanaikaisesti. Uudet ohjelmistot pyrkivät nopeuttamaan ja helpottamaan työnkulkua, jolloin artistille jää enemmän aikaa keskittyä lopputuloksen tarkasteluun ja yksityiskohtien hiomiseen. Yleisimpiä teksturointiohjelmia ovat muun muassa Adobe Photoshop, Quixel Suite 2 ja Allegorithmicin Substance-ohjelmat.

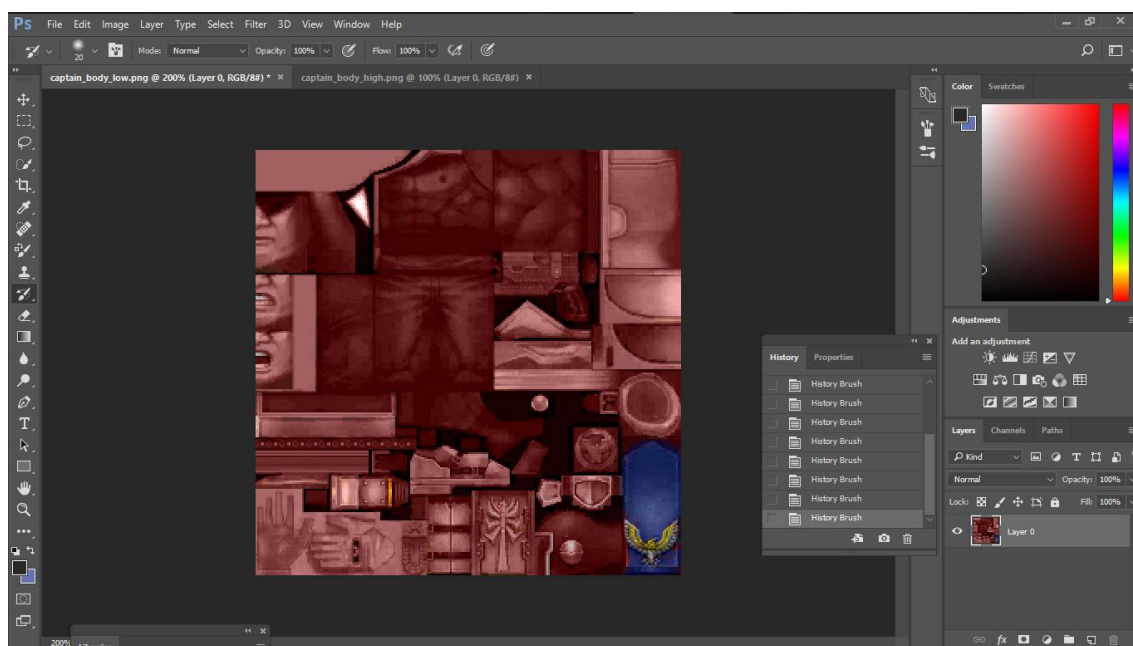
8.1 Adobe Photoshop

Adobe Photoshop on Adoben kehittämä kuvankäsittelyohjelma. Huolimatta maineestaan kuvanmuokkausohjelmana, Photoshop ei ole pelkästään kuvien retusointia ja käsittelyä varten. Sitä voidaan käyttää moneen muuhunkin tarkoitukseen, kuten luomaan omia tekstuurikarttoja. (Petrany 2014.)

Vaikka uudet 3D- ja PBR -teksturointiohjelmat ovat vähitellen korvaamassa perinteisiä kaksiulotteisia teksturointiin käytettyjä ohjelmia, Photoshop on säilyttänyt silti kannattajakuntansa. Photoshopin joustavuus ja edistykselliset sekoitus-, maalaus- ja peittämis työkalut tarjoavat edelleen vankan paketin teksturiartisteille, jotka eivät ole vielä sijoittaneet uusiin ohjelmistoihin tai jatkavat sen käyttöä näiden rinnalla. (Creative Bloq 2013.)

Photoshop tarjoaa yhä muutamia ominaisuuksia, joita kaikissa uusissa ohjelmistoissa ei vielä ole, kuten kyky säätää ja sekoittaa tasoja. Lisäksi käyttäjäystävällisyys ja ohjelmiston yleisyys tekevät ohjelman käyttämisestä hyvän ratkaisun, kun tekstuureita halutaan tehdä nopeasti ja vähällä vaivalla. (Creative Bloq 2013.)

Photoshopin ominaisuudet tarjoavat työkaluja ja tekniikoita, kuten säätötasoja, jotka mahdollistavat tuhoamattoman työnkulun, jonka avulla kuvia voidaan muokata tekemättä niihin peruuttamattomia muutoksia. Näiden avulla artisti pystyy säätämään tai jopa kokonaan kumoamaan tekemiään muokkauksia ilman liiallista uudelleen käsittelyä. Tämä tuhoamaton työnkulku on Photoshopin vahvimpia ominaisuuksia tekstuurikarttoja maalattaessa, jonka takia se pysyy olennaisena ohjelmana teksturiartistin työkalupakissa. (Creative Bloq 2013.) Kuva 34 esittää Photoshop-ohjelman käyttöliittymää.



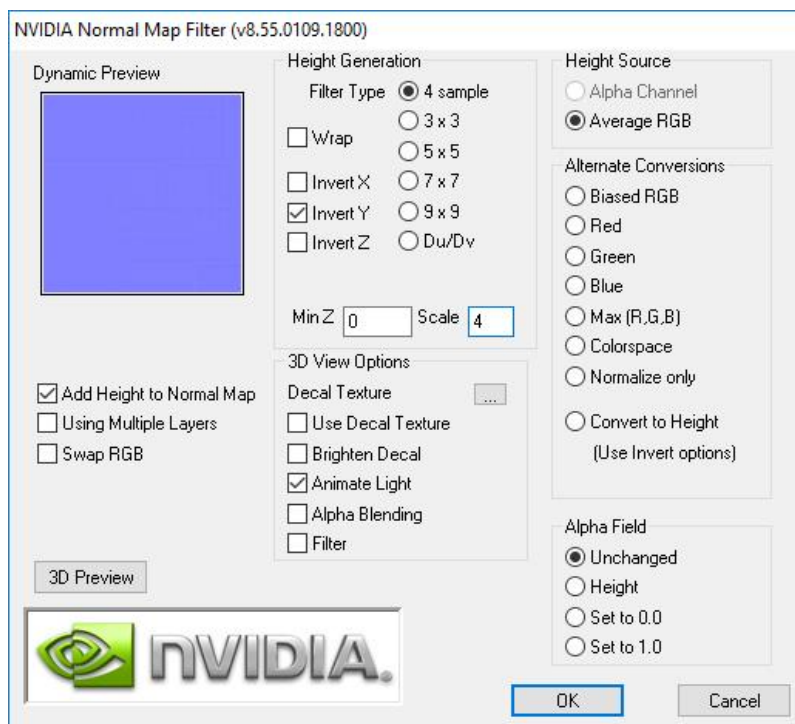
KUVA 34. Photoshop-kuvankäsittelyohjelman käyttöliittymä (GameBanana 2017)

Rescue 2 -pelin tuotannossa Photoshopia käytettiin pääasiallisena ohjelmana tekstuurien luonnissa. Käytetyistä tekstuurityypeistä diffuusio-, speulaarisuus- sekä normaali-kartat pystyttiin kaikki luomaan Photoshopin avulla, jolloin muita ohjelmia käytettiin lähinnä apuna niiden yksityiskohtien teksturoinnissa, joihin Photoshop ei ollut erikoistunut. Kaikki projektin artistit olivat harjaantuneita Photoshopin käytössä ja pelisarjan aikaisemman osan tekstuurit oltiin luotu Photoshopin avulla, joten ohjelman käyttäminen tekstuurien luonnissa oli luontainen valinta.

Photoshopin ohella markkinoilla on myös monia muita maksullisia sekä ilmaisia kaksiulotteisia kuvankäsittelyohjelmia, joita voidaan käyttää tekstuurien luonnissa samalla tavalla kuin Photoshopia. Ohjelmat eroavat toisistaan pääasiassa käyttöliittymältään ja ominaisuuksiltaan ja ohjelman valinta riippuu lähinnä teksturiartistin henkilökohtaisista mieltymyksistä.

8.2 NVIDIA Texture tools for Adobe Photoshop

NVIDIA Texture Tools for Adobe Photoshop on NVIDIAN kehittämä kokoelma tekstuurien luontiin tarkoitettuja työkaluja, joka toimii liitännäisenä Photoshop-kuvankäsittelyohjelmassa. Liitännäisen avulla on mahdollista pakata tekstuureita, muotoilla ympäristökarttoja sekä luoda normaali- ja MIP-karttoja. NVIDIA Normal Map Filter luo normaalikarttoja harmaasävyisistä korkeuskartoista. Suodattimen käyttöliittymä sisältää tehokkaan 3D-esikatselun sekä monipuoliset asetukset lopputuloksen säätämiseen. (NVIDIA Developer 2017.) Kuva 35 esittää NVIDIA Normal Map Filterin käyttöliittymää, jossa säädetään normaalikartan asetuksia.



KUVA 35. NVIDIA Normal Map Filter

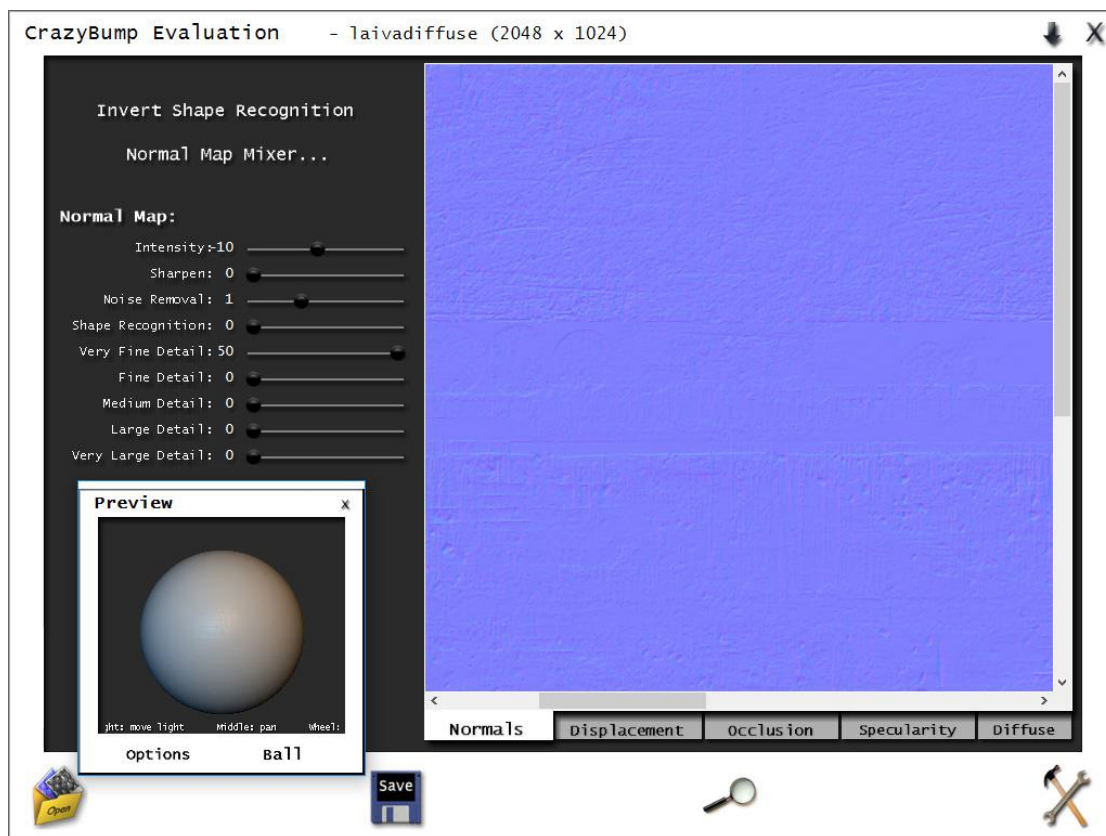
Rescue 2:n assettien teksturoinnissa NVIDIAN liitännäistä käytettiin normaalikarttojen luomisessa. Luominen tapahtui maalaamalla harmaasävyinen korkeuskartta diffuusio-
siokartan päälle, joka muutettiin suodattimen avulla normaalikartaksi ja tallennettiin lopulta omaksi tiedostokseen. Liitännäistä päädyttiin käyttämään sen nopean ja yksinkertaisen käyttöliittymän takia ja koska se on ilmainen.

8.3 CrazyBump

CrazyBump on nopea ja helppo työkalu normaalikarttojen luomiseen bittikarttakuvasta. Ohjelmalle syötetään kuvatiedosto, josta halutaan luoda korkeusarvot, jonka jälkeen ohjelma muuttaa kuvan harmaasävyiseksi. Harmaasävyinen kuva syötetään varjostimelle, jota käytetään lopputuloksen generoimisessa. (Goodsell 2017.)

Käyttäjällä on kuvan syöttämisen jälkeen mahdollisuus muokata yksityiskohtien suuruutta liukusäätimien avulla. CrazyBumpin oletusasetukset ovat harvoin sopivia käytettäväksi lopputuloksessa ja vaativat usein säätämistä optimaalisen lopputuloksen saavuttamiseksi. Kuvatiedostoa on kuitenkin syytä muokata ennen sen syöttämistä ohjelmalle, jotta normaalikartta voidaan generoida oikean valoinformaation perusteella. (Goodsell 2017.)

CrazyBump on ollut pitkään suosittu ohjelma normaalikarttojen luomisessa, mutta se alkaa vanhentua muiden samankaltaisten ohjelmien rinnalla (Slick 2017). Kuva 36 esittää CrazyBump-ohjelman käyttöliittymää, johon on syötetty kuvatiedosto normaalikartan luomista varten.



KUVA 36. CrazyBump-ohjelman käyttöliittymä

Rescue 2 -projektissa CrazyBumpia käytettiin normaalikarttojen luomisen apuna. Reaaliaikaisen 3D-esikatselun ja helppokäyttöisten liukusäätimien avulla diffuusiokartan pinnanmuodoista saatiin generoitua halutun syvyiset yksityiskohdat normaalikarttaan. Valokuvista luodut normaalikartat yhdistettiin Photoshopissa NVIDIA Texture Tools for Adobe Photoshop -liitännäisen avulla luotuun normaalikarttaan, jolloin 3D-mallin pinnalle saatiin isompien pinnanmuotojen lisäksi myös pienempiä yksityiskohtia.

Normaalikarttojen luomiseen on olemassa useita muitakin ohjelmia. Lisenssien hinnan ohella merkittävämpiä eroja ohjelmien välillä on käyttöliittymä ja niiden tuottamien tekstuurien laatu.

8.4 Quixel Suite 2

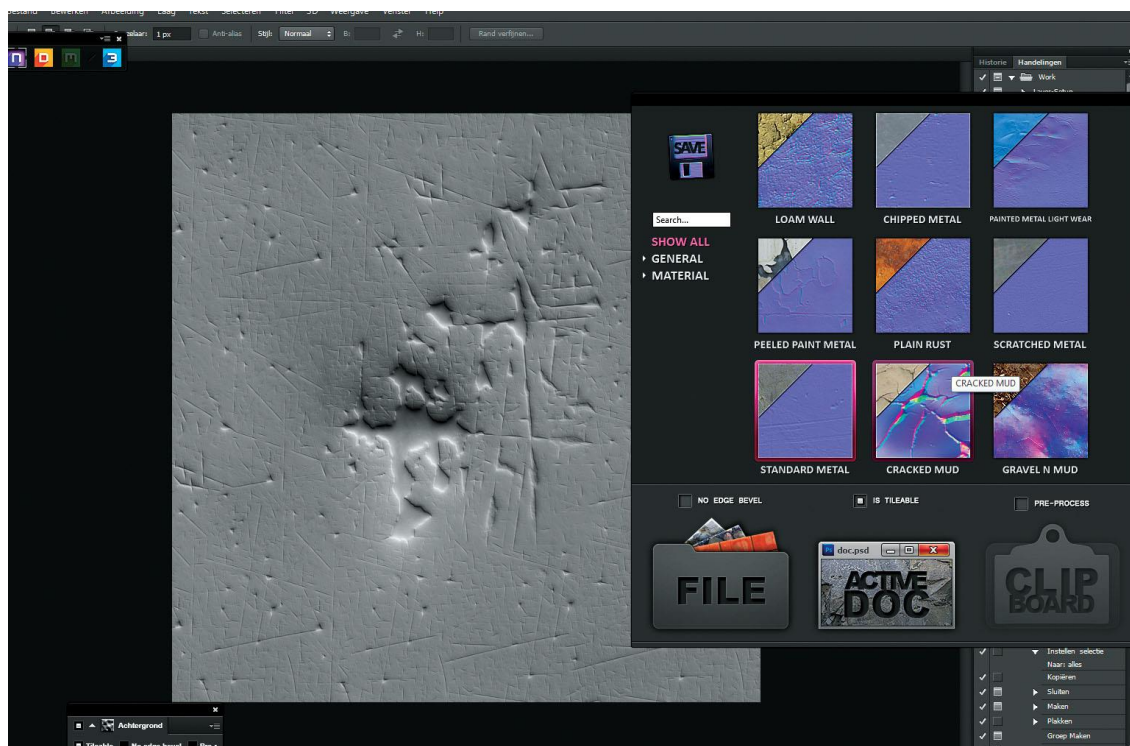
Quixel Suite 2 on Quixelin kehittämä skannattuihin materiaaleihin perustuva teksturointiohjelmistopaketti, joka toimii Adoben Photoshop-kuvankäsittelyohjelmassa. Quixel Suite 2 hyödyntää Photoshopin ominaisuuksia tekstuurien maalaamisessa ja se tukee

PBR-standardia. Ohjelmistopaketti on käyttäjäystävällinen ja sen käyttöliittymää voi muokata todella joustavasti. Quixel Suite 2 koostuu kolmesta ohjelmasta: DDO, 3DO ja NDO. (MotionMedia 2017.)

DDO -ohjelmaa käytetään tekstuurikarttojen luomiseen. Ohjelma sisältää laajan kirjaston skannattuja materiaaleja ja siveltimiä, joiden avulla tekstuurien maalaaminen on nopeaa ja lopputulokset yhdenmukaisia. (MotionMedia 2017.)

3DO on kolmiulotteinen maalaus ja esikatseluohjelma, jonka avulla 3D-malleja voidaan tarkastella Photoshopin käyttöliittymässä. Esikatselun lisäksi 3DO:n avulla voidaan maalata suoraan 3D-mallien pintaan Photoshopin omilla työkaluilla. (MotionMedia 2017.)

NDO -ohjelmaa käytetään normaalikarttojen luomiseen. NDO:n avulla jokaista Photoshopin työkalua, kuten vektoreita, valintatyökaluja ja mukautettuja siveltimiä on mahdollista käyttää normaalikarttojen reaaliaikaisessa veistämisessä. NDO pystyy myös tuottamaan normaalikarttoja valokuvista. (MotionMedia 2017.) Kuva 37 esittää NDO-ohjelman käyttöliittymää, joka toimii Photoshop-kuvankäsittelyohjelmassa.



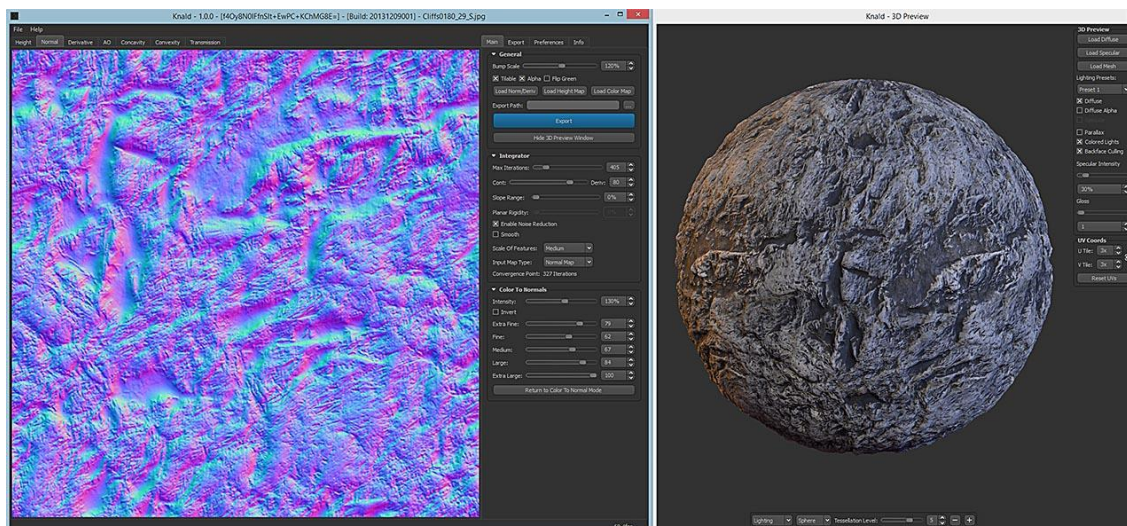
KUVA 37. NDO-ohjelman käyttöliittymä Photoshopissa (3D Artist 2016)

Rescue 2:n tuotannossa NDO:ta käytettiin joidenkin normaalikarttojen generoimiseen valokuvista. CrazyBumpia laajemmat asetukset yksityiskohtien säätämisessä ja laadukas kolmiulotteinen esikatselu toimivat hieman vanhentunutta CrazyBumpia paremmin, joten suoraan valokuvista generoitavat normaalit oli kannattavampaa toteuttaa NDO:lla.

8.5 Knald

Knald on Knald Technologiesin kehittämä uusi tekstuurikarttojen generointiin erikoistunut ohjelma, joka pystyy tuottamaan ympäristön okklusio-, kaviteetti- ja normaali-karttoja valokuvien ja korkeuskarttojen perusteella. Knald on yksi parhaimmista ohjelmista normaalikarttojen luomisessa. (Slick 2017.)

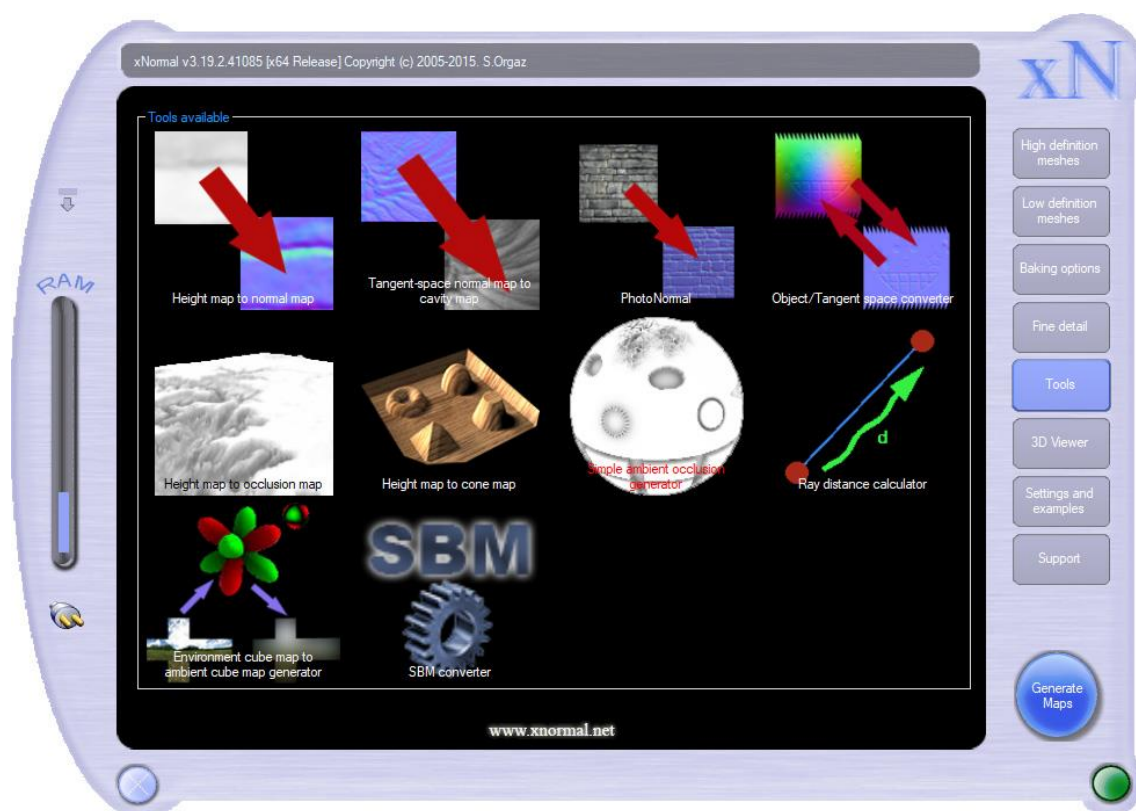
Karttojen generointi Knaldilla nopeaa ja ohjelma sisältää yhden parhaista 3D-esikatselunäkymistä lopputuloksen tarkastelua varten. (Slick 2017.) Kuva 38 esittää Knald-ohjelman käyttöliittymää, jossa vasemmalla puolella on generoitava normaali-kartta ja oikealla ohjelman esikatseluikkuna.



KUVA 38. Knald-ohjelman käyttöliittymä (obedient echo 2015)

8.6 XNormal

XNormal on Santiago Orgazin kehittämä suosittu ja laajasti käytössä oleva ohjelma, jonka avulla korkearesoluutioisen 3D-mallin pinnasta pystytään generoimaan normaali-karttoja. Ohjelma on ilmainen ja todella moni tekstuuriartisti käyttää sitä työssään. XNormal on erittäin tehokas normaalikarttojen leipomiseen ja se pystyy tuottamaan myös laadukkaita ympäristön okklusiokarttoja. Vaikka XNormal on ilmainen, pystyy se tuottamaan yhtä laadukasta jälkeä kuin maksulliset kilpailijansa. (Slick 2017.) Kuva 39 esittää XNormal-ohjelman käyttöliittymää.



KUVA 39. XNormalin käyttöliittymä

8.7 Substance Designer, Substance Painter ja Substance B2M

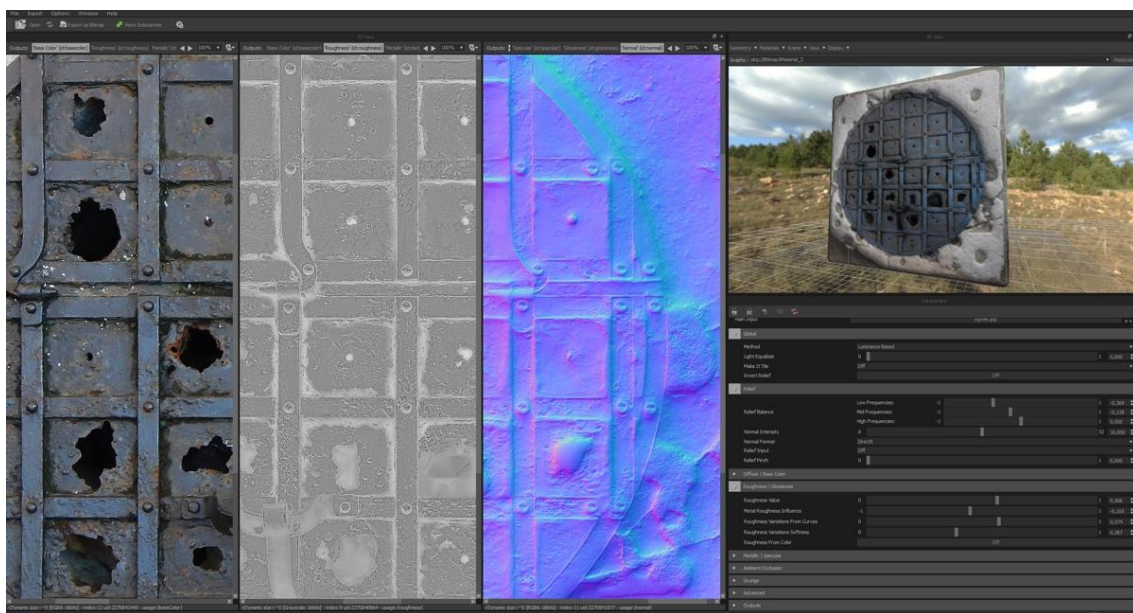
Substance Designer on Allegorithmicin kehittämä toimintorikas teksturointityökalu, joka käyttää solmupohjaista graafista käyttöliittymää. Substance Designer on erityisen hyvä laatoittuvien tekstuurien luonnissa. Solmupohjaisen käyttöliittymän ansiosta teks-

tuurien muokkaaminen on nopeaa ja kokenut artisti pystyy saamaan hetkessä aikaan näyttäviä tuloksia. (Slick 2017.)

Substance Painter on 3D-maalausohjelma, jolla voi teksturoida, renderöidä ja jakaa omia tuotoksiaan. 3DOn tapaan Substance Painter tukee PBR-tekniikkaa. Ohjelma sisältää siveltimiä joiden peittävyyttä voidaan hallita, proseduraalisia efektejä, peittomaskeja sekä muita Photoshopistakin tuttuja työkaluja. (Allegorithmic 2017.)

Substance B2M-ohjelman avulla on mahdollista luoda kaikki PBR-varjostimien tarvitsemat tekstuurikarttatyypit: perusväri-, normaali-, metalli-, karheus- sekä ympäristön okklusiokartan. Ohjelma sisältää laajan valikoiman asetuksia, joiden avulla tekstuurien yksityiskohtia voidaan säätää. B2M-ohjelma pystyy myös poistamaan kuvista automaattisesti saumakohtia, joten laatoittuvien tekstuurien generoiminen on sillä helppoa. (Allegorithmic 2017.)

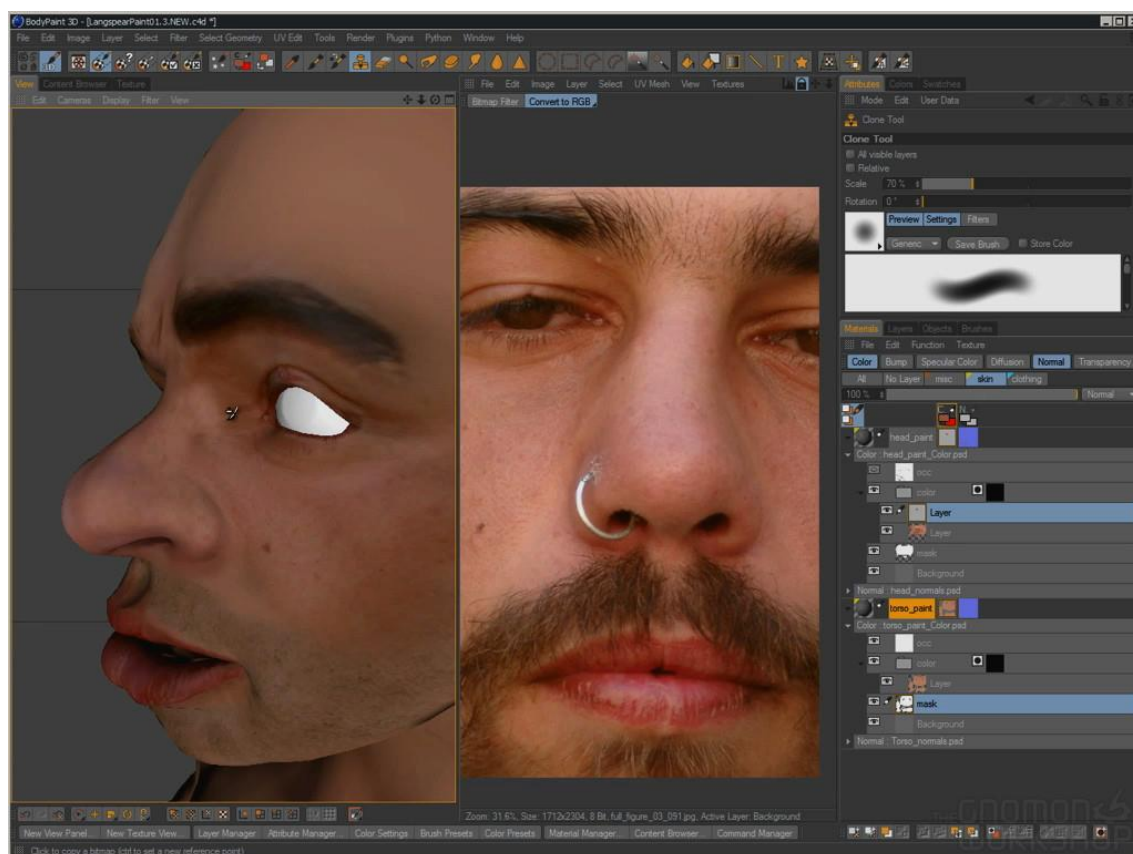
Kuva 40 esittää Substance B2M -ohjelman käyttöliittymää. Vasemmalla ovat ohjelman generoimat tekstuurikartat ja oikealla 3D-mallin esikatseluikkuna sekä tekstuurien hienosäätöön käytettävät liukusäätimet.



KUVA 40. Substance B2M -ohjelman käyttöliittymä (Allegorithmic 2017)

8.8 BodyPaint 3D

BodyPaint 3D on Maxonin kehittämä 3D-maalausohjelma, joka sisältää joitakin melko uniikkeja ominaisuuksia, kuten monikanavamaalauksen ja mahdollisuuden käyttää Photoshopin siveltimiä kolmiulotteisessa työtilassa. BodyPaint 3D tarjoaa kehittyneempiä työkaluja, kuin useimmat mallinnusohjelmiin sisäänrakennetut 3D-maalautustyökalut. (Slick 2017.) Kuva 41 esittää BodyPaint 3D -ohjelman käyttöliittymää, jossa maalataan ihmiskasvoja.



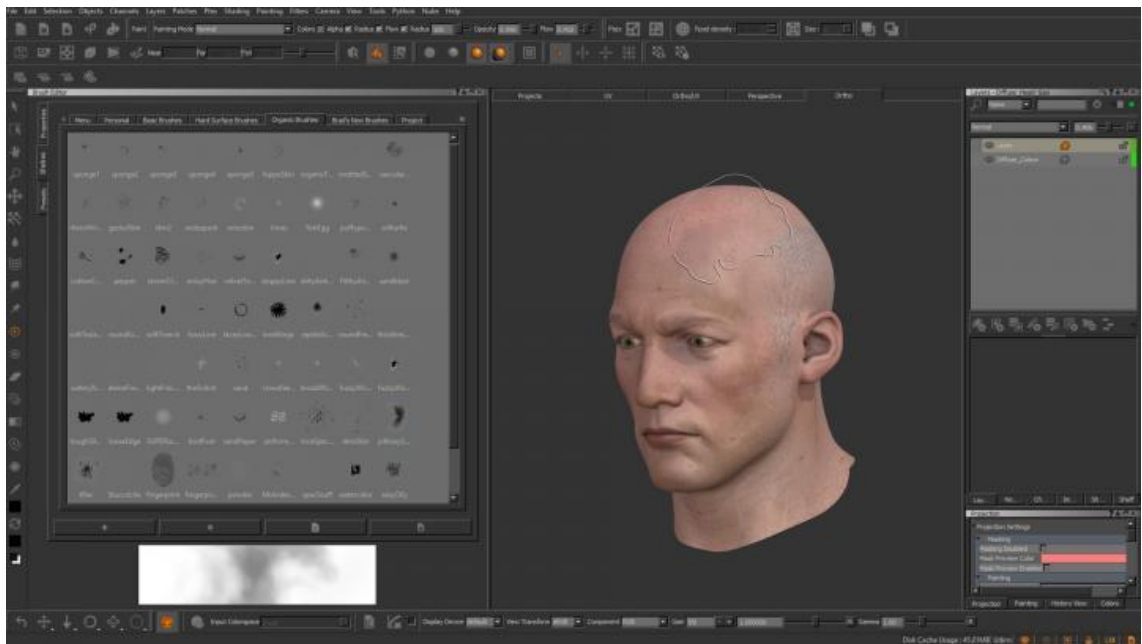
KUVA 41 BodyPaint 3D -ohjelman käyttöliittymä (The Gnomon Workshop 2017)

8.9 Mari

Mari on Foundryn kehittämä 3D-maalausohjelma, joka on suunniteltu käytettäväksi huippuluokan elokuva-, televisio- ja pelituotannoissa (Slick 2017). Marin avulla on mahdollista luoda todella korkean resoluution tekstuurikarttoja. Ohjelma sisältää mah-

dollisuuden käyttää sekä Marin omia, että Photoshopin siveltimiä, eri sekoitusmuotoja, tasoja ja peittomaskeja. (Foundry 2017.)

Ohjelma tukee useita UV-tekniikoita ja 3D-mallien tahkoille on myös mahdollista maalata ilman UV-kartoitusta. Tahkojen tekstuureiden tarkkuutta on myös mahdollista säätää erikseen. (Foundry 2017.) Yksityisille artisteille ohjelman melko kallis lisenssi saattaa tosin olla ongelmallinen (Slick 2017). Kuva 42 esittää Mari-ohjelman käyttöliittymää ja siveltimiä, joita käytetään 3D-mallien teksturoinnissa.



KUVA 42. Mari-ohjelman käyttöliittymä (Foundry 2017)

9 TEKSTUURIKARTAN LUOMINEN RESCUE 2 -PELISSÄ

Tavat joilla artisti työskentelee, työkalut mitä hän käyttää ja järjestys, joissa työn vaiheita tehdään, tunnetaan peligrafiikan tuottamisesta puhuttaessa termillä "pipeline", joka tarkoittaa prosessien joukkoa. Itsenäisessä työssä ja projektityöskentelyssä pipeline pyritään suunnittelemaan niin, että työskentely etenee mahdollisimman sulavasti. (Gahan 2013, 159.)

Rescue 2 -projektissa työnkulku aloitettiin suunnittelemalla tarkkaan, mitä asetteja pelissä tarvitaan. Suunnittelun jälkeen asetin tekoa varten etsittiin sopivat mallikuvat ja tarvittaessa projektin Creative Director piirsi asetin konseptikuvan tukemaan valokuvaesimerkkejä. Peliobjektin ulkonäön ollessa selvillä artisti aloitti mallintamaan asettia referenssien perusteella. Mallinnusprosessin aikana asettia testattiin pelimoottorissa ja sitä iteroitiin sen mukaan, jos mallissa oli korjattavaa.

Mallinnusprosessin ollessa lähes valmis, malli tarkastutettiin vielä lisämuutosten varalta, ennen kuin varsinaista UV-kartoitusta alettiin tehdä. UV-kartoitus pyrittiin tekemään nopeasti mutta tehokkaasti, jotta teksturointiprosessi sujuisi vaivattomasti. Itse teksturointi tapahtui aikaisemmin valittuja referenssikuvia käyttämällä. Teksturoinnin edetessä Creative Director antoi kehitysehdotuksia, jotta valmiista tekstuurista saataisiin laatuvaatimuksia vastaava. Valmis 3D-malli ja tekstuurikartat vietiin pelimoottoriin ja muutokset tallennettiin versionhallintaan, jonka kautta kenttäsuunnittelija pystyi käyttämään niitä pelimaailmaa rakentaessaan.

Projektissa teksturointiin käytetyt työkalut olivat Adobe Photoshop-kuvankäsittelyohjelma, NVIDIA Texture Tools for Adobe Photoshop -liitännäinen sekä CrazyBump ja Quixel Suiten NDO -ohjelmat. 3D-mallinnuksessa käytettiin Autodeskin 3d Studio Max -ohjelmaa.

Projektissa ei ollut käytössä varsinaista tyyliopasta, joten tekstuurien yhteneväisen tyylin ylläpitämisessä apuna käytettiin Creative Directorin ohjeita, esituotannossa luotua markkinointimateriaalia sekä pelisarjan aikaisemman osan grafiikoiden tyyliä.

Tässä luvussa käydään läpi erään pelissä käytetyn assetin teksturointiprosessi. Teksturoinnissa käytettiin Photoshop-kuvankäsittelyohjelman englanninkielistä versiota, joten termistö on suomennettu, mutta selvyys vuoksi niiden englanninkieliset termit ovat käännetty sitä mukaa, kun suomenkieliset termit mainitaan.

Tehtävä kahdeksan sijoittui satama-alueelle, jonka laituriin ankkuroidun rahtilaivan kannella on sattunut konttipalo. Tehtävänä oli luoda rahtilaivan 3D-malli sekä teksturoida valmis 3D-malli. Seuraavissa alaluvuissa selostetaan, miten rahtilaivan teksturointiprosessi tapahtui.

9.1 Dokumentin luominen ja alkuvalmistelut

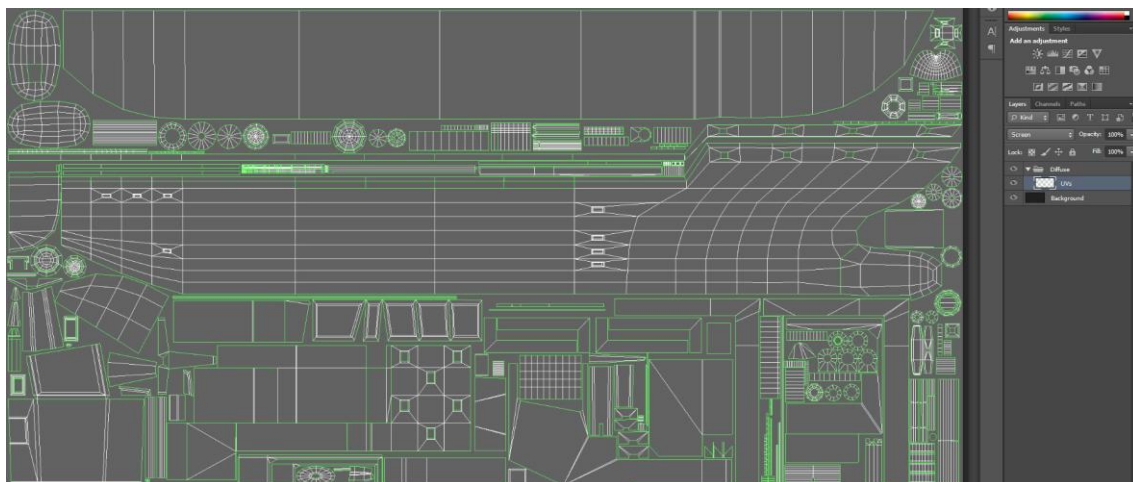
Teksturointiprosessi aloitettiin päättämällä tekstuurikartalle sopiva resoluutio. Koska kyseessä oli niin sanottu "hero asset", eli iso ja näyttävä objekti, joka toimi huomion keskipisteenä satamassa, oli tärkeää käyttää tarpeeksi isoa resoluutiota, jotta yksityiskohdat näkyisivät laivaa tarkasteltaessa selkeästi. Tämän assetin teksturoimisessa päätettiin käyttää 2048 x 1024 pikselin resoluutiota. Teksturoinnin nopeuttamiseksi laivasta tehtiin symmetrinen ja sen toinen puoli peilattiin UV-editorissa toisen puolen päälle, jolloin molempien puolikkaiden oli mahdollista käyttää samaa tekstuuria. Ajan säästämisen lisäksi tällä tavoin saatiin maksimoitua tekstuurin resoluutio tahkojen pinnalla.

Photoshopissa luotiin uusi dokumentti, jolle annettiin päätetyt mitat. Teksturoinnin apuna käytettiin 3D-mallin UV-saarekkeita. Käyttämällä 3ds Maxin UV-editorista löytyvää Render UVW Template -toimintoa, objektin UV-piirrustukset kopiottiin leikepöydälle. Oli tärkeää, ettei objektin geometriaan tehty enää teksturoinnin aloittamisen jälkeen huomattavia muutoksia, sillä ne olisivat rikkoneet UV-kartoituksen, joka olisi aiheuttanut vääristymistä objektin pinnassa.

Leikepöydällä oleva UVkartta liitettiin Photoshopissa luodun dokumentin päälle. Tason (Layer) sekoitusmuodoksi (Blending Mode) valittiin rasteri (Screen) ja peittävyys (Opacity) määräksi noin 50 prosenttia. Tällä tavoin UV-piirrustukset eivät olleet tasojen tiellä näkemistä vaikeuttamassa, mutta säilyttivät silti luettavuuden. Tämä taso pidettiin päällimmäisenä tasopinossa (Layer Stack) koko työskentelyn ajan.

UV-piirustusten alle luotiin uusi taso, joka värjättiin keskiharmaaksi (kuva 43). Jokaiselle jatkossa luotavalle elementille oli tärkeää luoda oma tasonsa, jotta tekstuuria voitiin tarvittaessa muokata joustavammin. Sisältöä vastaavan nimen antaminen tasoille helpotti dokumentin ylläpidettävyyttä ja nopeutti työskentelyä, kun oikeaa tasoa ei tarvinnut etsiä muiden joukosta.

Tasojen määrä valmiissa tekstuurissa voi olla useita kymmeniä ja nimeämättömät tasot vaikeuttavat dokumentin muokkaamista, varsinkin jos toinen artisti joutuu tekemään tekstuuriin muokkauksia myöhemmin.



KUVA 43. Tekstuurin UV-kartta

9.2 Pohjavärialueiden luominen

Taustan päälle luotiin uudet tasot laivan pohjavärejä varten. Samanväriset alueet oli hyvä tehdä omille tasoilleen. Referenssikuvia tutkimalla voitiin nähdä, mitä värejä laivassa käytettiin. Valintatyökalujen avulla UV-kartassa näkyvät 3D-mallin osat täytettiin omilla väreillään. Monikulmiolasso (Polygonal Lasso) käyttämällä alueet saatiin rajattua tarkasti. Eriväristen alueiden ympärille oli hyvä jättää muutaman pikselin reunat, jotta taustaväri ei vuotaisi niiden päälle. Laivan runko täytettiin pääosin tummansinisellä, punaisella, valkoisella ja vihreällä värillä (kuva 44).

Kun siniset osat olivat värjätty, siirryttiin rajaamaan muita alueita omille tasoilleen. Jokaiselle materiaalille, jotka käyttivät samaa väriä, luotiin oma tasonsa. Pohjaväreinä

käytettiin tummia sävyjä. Täysin valkoisia, keskiharmaita ja mustia värejä vältettiin, sillä ne ovat harvinaisia luonnossa.



KUVA 44. Täytetyt pohjavärialueet

9.3 Varjojen lisääminen

Kun pohjavärialueet oli rajattu, voitiin niiden päälle lisätä ympäristön okklusiokartasta saatavat varjot. Objektin okklusiokartta saatiin piirrettyä 3ds Maxin ”Render To Texture” -toiminnolla. Ikkunasta voitiin valita tiedostopolku, tiedoston formaatti sekä tekstuurin resoluutio. Okklusiokartan resoluutioksi määritettiin sama kuin tekstuurikartallakin, sillä pienemmän kuvan käyttäminen olisi tehnyt varjoista suttuisia. Valmis okklusiokartta avattiin Photoshopissa ja liitettiin pohjavärien päälle (kuva 45). Tason sekoitusmuodoksi valittiin kertova (Multiply), jolloin se päästi taustalla olevat värit läpi, jättäen tummat alueet näkyville. Tason peittävyttä säädettiin sen mukaan, kuinka tummalta se mallin päällä näyttää. Tasosta voidaan tehdä myös toinen kopio, jos jotkin alueet näyttävät liian vaaleilta. Joskus okklusiokarttaan on piirtynyt virheellisiä varjostuksia ja nämä pyyhittiin tasosta pois ennen teksturoinnin jatkamista.

Joissain tapauksissa oli tarpeen maalata myös varjoja tekstuuriin itse, kun okklusiokarttaan niitä ei ollut piirtynyt ja kun eri alueita objektin pinnassa haluttiin korostaa lisää. Varjojen maalaamiseen käytettiin pehmeää pyöreää sivellintä, (Soft Round Brush) jonka virtaus (Flow) on asetettu noin 30 prosenttiin ja peittävyys sataan prosenttiin. Käyttämällä isoa sivellinkokoa saatiin aikaan pehmeämpi lopputulos. Kohtia, joissa

muut pinnanmuodot estivät valoa, korostettiin. Samoin pinnasta nousevia elementtejä ns. istutettiin, eli varjokohtia maalattiin niiden ympärille. Näin pintaan saatiin syvyyttä ja kappaleille yhteenkuuluvuutta muuhun geometriaan. Maalatut varjot sisällytettiin omalle tasolle, jotta niiden peittävyttä ja muotoa voitiin tarvittaessa hallita.



KUVA 45. Ympäristön okklusiokartta ja maalatut varjot

Mallin pintaan jäljiteltiin pullistumia ja koloja, joita varsinaisessa geometriassa ei ollut. Maalaamalla varjoja sinne, minne ne luonnollisesti lankeisivat, saatiin aikaan realistisen näköisiä yksityiskohtia ilman lisägeometriaa (kuva 45). Pehmeällä pyyhkeumilla (Eraser) liiallisia varjoja pyyhittiin pois. Kuten pohjavärien kanssa, täysin mustia alueita vältettiin, koska geometria voi altistua pelimoottorissa valolle.

9.4 Väriliukujen lisääminen

Tummilla väriliukuilla saatiin lisättyä syvyyttä pystysuorille pinnoille ja ohjattua pelaajan katsetta huomionarvoisiin kohtiin. Pohjavärien päälle luotiin uusi taso, kuten varjoja maalatessakin. Pehmeällä pensselillä ja väriliukutyökalulla (Gradient Tool) luotiin musta väriliuku ylöspäin laivan kylkeä, hyttien seiniä ja kannen viistoa ramppia pitkin (kuva 46). Jotta ylimennyttä väriliukua ei tarvinnut pyyhkiä erikseen pois, sitä leikattiin pohjaväritasojen sisällöllä, käyttämällä leikkausmaskia (Clipping Mask). Valitsemalla väriliuku tasopinosta ja painamalla hiiren oikeaa painiketta, valittiin Create Clipping Mask -toiminto, jolloin allaoleva taso leikkasi ylem্পää tasoa. Tämä on hyödyllinen ominai-

suus, jota hyödynnettiin seuraavissa vaiheissa. Leikatun väriliu'un peittävyttä säädettiin niin alhaiseksi, niin että se oli nähtävissä, muttei erottunut pinnasta liikaa.



KUVA 46. Suurille pinnoille lisätyt väriliu'ut

9.5 Saumojen maalaaminen

Joissakin tilanteissa objektien pinnoilla saattaa esiintyä niin pieniä yksityiskohtia, että niiden tekeminen 3D-malliin olisi turhaa. Tällaisia yksityiskohtia voitiin jäljitellä teksturiin piirtämällä ne siihen suoraan. Rahtilaivan kyljissä olevat saumat olivat tällaisia yksityiskohtia. Varjojen päälle luotiin uusi taso, jolle saumat piirrettiin (kuva 47). Pensselin peittävyudeksi ja virtaukseksi asetettiin sata prosenttia ja väriksi valittiin mahdollisimman tumma väri. Koska saumat olivat todella pieniä, pensselin kooksi valitaan yksi pikseli. Tason peittävyttä hallittiin sen mukaan, kuinka hyvin saumat erottuivat pinnasta.



KUVA 47. Laivan kylkiin ja kannelle lisätyt saumat

9.6 Korostuksien maalaaminen

Aikaisemmin maalattuja syvennyksiä ja varjoja vahvistettiin piirtämällä korostuksia niihin kohtiin, joihin valo osuisi. UVs-tason alle luotiin uusi taso, jonka sekoitusmuodoksi valittiin sulauttava (Overlay). Pensselin kooksi valittiin yksi pikseli, jolla 3D-mallissa olevien terävien reunojen kohdalle maalattiin valkoisella värillä. Oikean geometrian reunoille piirtämisen lisäksi myös jäljiteltujen yksityiskohtien syvyysvaikutelmaa vahvistettiin maalaamalla niiden reunoille korostukset. Tässä tapauksessa laivan kyljissä olevien saumojen reunoille maalattiin raidat (kuva 48).

Uuden tason päälle luotiin toinen taso pehmeämpiä korostuksia varten. Sen sekoitusmuodoksi valittiin myös sulauttava. Pehmeitä korostuksia maalattiin pyöreämpiin kohtiin, kuten pelastusveneen kylkeen ja pelastusrenkaisiin. Pehmeiden korostusten maalaamiseen käytettiin isompaa ja pehmeämpää pensseliä ja niiden muotoa hienosäädettiin pehmeällä pyyhekumilla. Toisensa peittävät, isommat korostukset luotiin erillisille tasolle, jotta niiden muokaaminen erikseen olisi helpompaa.



KUVA 48. Geometrian reunoille ja saumojen ympärille maalatut korostukset

9.7 Yksityiskohdat ja sävymuutokset

Pintojen elävöittämiseksi tekstuuriin lisättiin yksityiskohtia, kuten rungossa olevat pultit ja merkinnät vesirajan yläpuolella (kuva 49). Referenssikuvien avulla saatiin hyvä käsitys siitä, millaisia yksityiskohtia tekstuuriin haluttiin lisätä. Osa yksityiskohdistista maalattiin itse ja osaan käytettiin valokuvista poimittuja osia.

Sävymuutoksien avulla samaa materiaalia olevat alueet saatiin erottumaan toisistaan ja tekstuuriin saatiin vaihtelua. Tutkalaitteiden ja muiden valkoisten osien kirkkautta säädettiin niin, että ne erottuivat alla olevista elementeistä.



KUVA 49. Laivan kannen, kyljen ja pelastuslaivan päälle lisätyt yksityiskohdat

9.8 Lian lisääminen

Usein mikään pinta, huolimatta kuinka puhtaan tai monotonisen näköinen tahansa, ei ole täysin puhdas tai ilman pintarakennetta. Tämän takia tekstuuriin lisättiin erilaisia kulumia, valumia sekä muita pintarakennetta kuvastavia yksityiskohtia aitouden lisäämiseksi ja monotonisuuden välttämiseksi. Referenssikuvia tarkastelemalla kartoitettiin, millaisia materiaaleja eri puolille laivaa haluttiin.

Koska laivan rungossa käytettiin pääasiassa metallia, sitä varten etsittiin sopiva metallinen tekstuuri, jota laivan pinnoilla käytettiin. Textures.com on hyvä sivusto, josta on mahdollista ladata ilmaiseksi tekijänoikeusvapaita tekstuureita. Laivan runkoa varten sivustolta etsittiin laatoittuva metallia esittävä kuva sekä kuvat ruostetta ja valumia varten. Kuvat liitettiin rungon pohjaväritason kohdalle kaikkien tasojen ylle ja niiden sekoitustasoksi valittiin sulauttava (kuva 50). Sekoitusmuotoa vaihdettiin sen mukaan, miten se vaikutti tason ulkonäköön. Pohjaväritason avulla lian ja materiaalien yli menevät osat saatiin leikattua pois. Tasot- (Levels) tai kirkkaus/kontrasti (Brightness/Contrast) -toiminnoilla metallin ja valumien tummuutta ja näkyvyyttä vähennettiin sen mukaan, miten ne toimivat alla olevien tasojen päällä.



KUVA 50. Erityyppiset metallit sekä kuluma- ja valumajäljet lisättynä pohjavärien päälle

Loput alueet laivasta käytiin läpi tarkastelemalla referenssikuvia ja päällystämällä ne asiaankuuluvilla pintamateriaaleilla.

9.9 Tekstien lisääminen

Laivan runkoon haluttiin lisätä yksityiskohtien ja kuluman lisäksi laivan nimet sekä varoitustekstit. Koska laivan toinen puolikas peilattiin UV-kartassa, tekstejä ei voitu suoraan lisätä tekstuuriin, muuten teksti olisi näkynyt peilattuna toisella puolella laivaa. Tämä ratkaistiin luomalla 3D-mallinnusohjelmassa tahkot niille paikoille, mihin tekstin haluttiin tulevan. Laivan kyljissä olevat tekstit saatiin tehtyä kopioimalla tahkot suoraan laivan kyljestä uudeksi objektiksi. Photoshopissa luotiin uusi tekstuuri runkoon tulevia tekstejä varten ja se tallennettiin .PNG-muodossa, jotta tekstin ulkopuoliset alueet saatiin läpinäkyväksi (kuva 51). UV-kartoituksen avulla tekstit saatiin heijastettua tahkoissa oikeille paikoilleen.



KUVA 51. Laivan nimien tekstuurikartta ja tahkot, joissa tekstuurikarttaa käytettiin

9.10 Lopputuloksen tarkastelu ja tekstuurin tallentaminen

Teksturointiprosessin ohessa teksturoitava 3D-malli renderöitiin säännöllisin väliajoin, jotta saatiin käsitys siitä, miltä tekstuuri näyttäisi pelissä. Tällöin saatiin paikannettua mahdolliset ongelmakohdat ennen peliin viemistä ja säästyttiin tekstuurin korjaamiselta myöhemmin. Teksturoitava renderöitiin niistä kuvakulmista ja etäisyyksiltä, miltä se tultaisiin pelissä näkemään. Näin saatiin paras käsitys tekstuurien toimivuudesta.

Valmista tekstuurikarttaa terävöitettiin lopuksi. Näin tekstuurista saatiin pois ylimääräinen sameus. Käyttämällä epäterävä maski (Unsharp Mask) -toimintoa, terävyyden määrää saatiin säädettyä sen verran, kun oli tarpeellista.

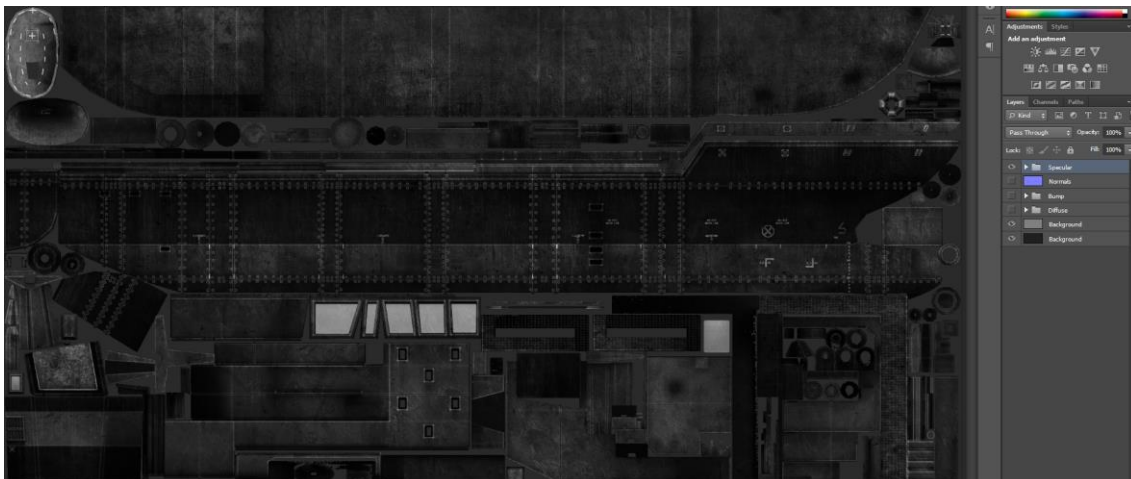
Valmis tekstuuri tallennettiin projektissa käytettävään bittikarttamuotoon ja nimettiin objektin ja tekstuurikarttatypin nimellä. Tässä tapauksessa kuvan nimeksi tuli ”CargoShip_diffuse”. Nimestä voitiin tällöin lukea missä 3D-mallissa sitä käytettiin ja mikä tyyppinen tekstuurikartta oli. Rescue 2 -projektissa diffuusiotekstuurien pääasiallisina tiedostotyyppinä käytettiin .TGA- ja .PNG -tiedostotyyppejä. Tässä tekstuurissa päätettiin käyttää .PNG-muotoa, joten tallennettaessa pudotusvalikosta valittiin .PNG. Pakkaustasoa kysyttäessä valittiin ”None / Fast”, jolloin kuvan laatu ei heikentynyt pakattaessa liikaa.

9.11 Spekulaarisuuskartan luominen

Jotta spekulaarisuuskarttaa voitiin aloittaa tekemään, valmiista diffuusiokartasta tallennettiin uusi kopio. Spekulaarisuuskartat ovat harmaasävyisiä, joten dokumentti haalistettiin käyttämällä haalista-toimintoa (Desaturate).

Spekulaarisuuskartassa vaaleat alueet heijastavat eniten valoa ja tummat taas vähemmän. Referenssikuvia tutkimalla voitiin päätellä, kuinka kirkkaita eri materiaalien heijastuksista haluttiin. Laivan rungossa oleva kulunut, maalattu metalli heijastaa valoa melko vähän, kun taas muovinen pelastusvene ja laivan ikkunat huomattavasti enemmän. Tällöin metalliosista tehtiin tummempia ja muovi- sekä lasiosista vaaleampia.

Sävy/kylläisyys (Hue/Saturation), tasot- ja kirkkaus/kontrasti -toiminnoilla tasojen vaaleutta saatiin muutettua nopeasti. Pohjavärien kirkkautta säädettiin sen mukaan, miltä niiden kuuluisi näyttää valon osuessa niihin (kuva 52). Korostuksien heijastusta lisättiin nostamalla tasojen peittävyyttä.



KUVA 52. Valmis spekulaisuuskartta

Spekulaisuuskartan toimivuutta testattiin ennen pelimoottoriin viemistä 3ds Max -ohjelmassa, jotta saatiin kuva siitä, onko kartta halutun kaltainen. Ohjelmassa 3D-mallin viereen luotiin valonlähde ja laivasta renderöitiin kuva, jolloin nähtiin, millaisia heijastukset olivat. Kun spekulaisuuskartta oli valmis, se tallennettiin .PNG -muodossa ja sen nimeksi annettiin ”CargoShip_specular”

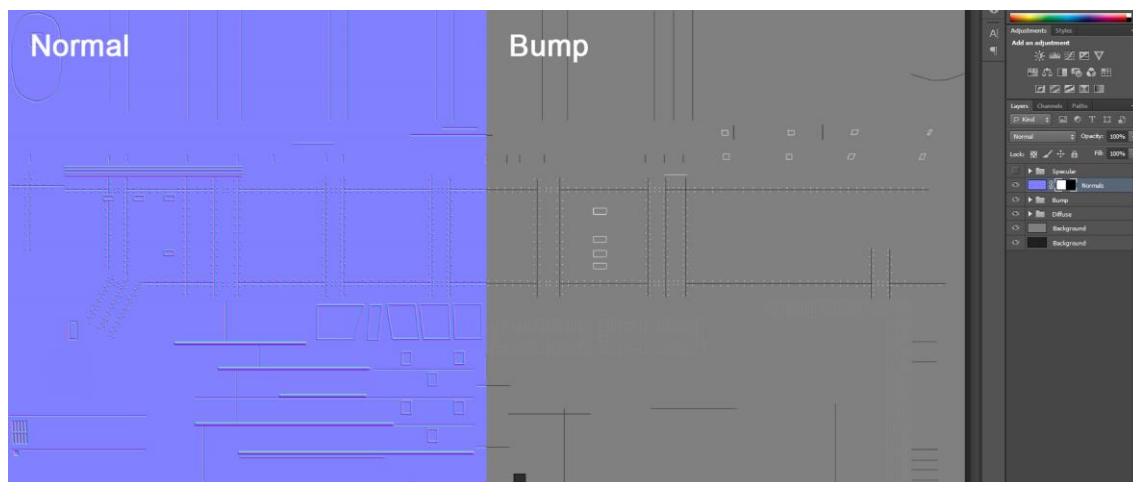
9.12 Normaalikartan luominen

Rahtilaivalle haluttiin tehdä normaalikartta, jotta diffuusiotekstuuriin jäljitellyt töyssyt ja syvennykset näyttäisivät aidommilta. Jotta normaalikartta saatiin generoitua, oli kyseiset yksityiskohdat tehtävä ensin kohoumakartan muotoon, jolloin ne maalattiin eri harmaasävyillä sen mukaan, kuinka ylös tai alas ne työntyvät pinnasta.

Diffuusiotekstuurin tasopinon päälle luotiin uusi ryhmä (Group), jolle annettiin nimeksi ”Bump”. Ryhmän sisälle luotiin uusi taso, joka täytettiin keskiharmaalla. Tämän tason päälle luotiin tasot, joihin kohoumat ja syvennykset tulivat. Korkeuskohdat, kuten saumat ja pultit kopioitiin suoraan aikaisemmin tehdyistä tasoista ja niiden kirkkautta muutettiin sen mukaan, kuinka paljon niiden haluttiin tulevan ulos pinnasta. Saumat muutettiin tummanharmaiksi, ja laivan kyljessä olevat pultit vaaleanharmaiksi (kuva 53).

Kun kaikki yksityiskohdat, jotka haluttiin siirtää normaalikarttaan, oli piirretty, voitiin kohoumakartassa olevat tasot yhdistää uudeksi tasoksi painamalla näppäimistöä Ctrl +

Alt + Shift + E. Yhdistetylle tasolle valittiin Filter-pudotusvalikon NVIDIA Tools -kohdasta NormalMapFilter -toiminto. Liitännäisen ikkunasta säädettiin lopputulokselle halutut arvot. OK-painiketta painamalla suodatin generoi harmaasävyisestä kohoumakartasta normaalikartan (kuva 53).



KUVA 53. Kohoumakartan ja siitä luodun normaalikartan vertailu

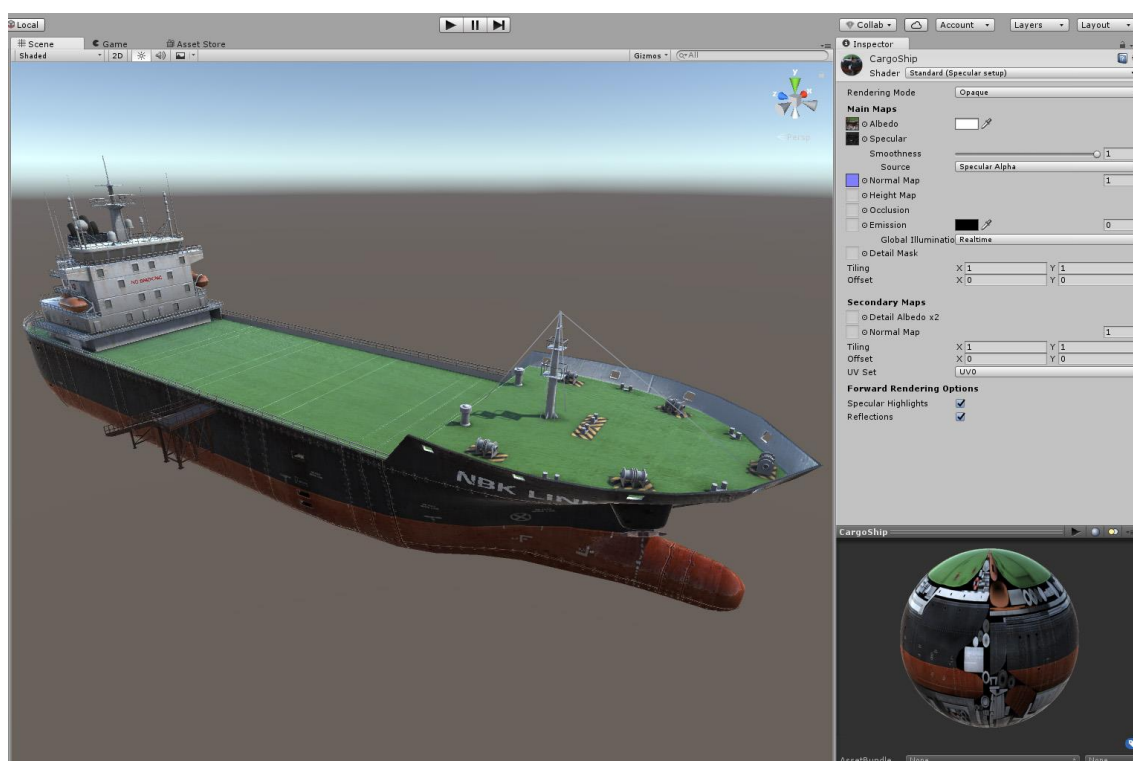
Jotta laivan pintaan oltaisiin saatu lisää realismin tuntua, CrazyBump-ohjelman avulla päätettiin luoda normaalikarttaan myös pienempiä yksityiskohtia. Tämä tapahtui generoimalla toinen normaalikartta diffuusiotekstuurissa olevista kuluma- ja materiaalitaistoista. Generoitavasta kuvasta oli tärkeää poistaa kaikki ylimääräinen valoinformaatio, sekä yksityiskohdat, jotka eivät kommunikoineet pinnanmuodossa olevia kolhuja tai pullistumia, muuten ohjelma generoi näistä virheellisiä syvyystietoja normaalikarttaan. Nämä tasot piilotettiin ja kuvasta tallennettiin pelkistetty versio, joka syötettiin ohjelmalle. CrazyBumpin liukusäätimien ja esikatseluikkunan avulla pinnanmuodoista saatiin luotua tarkat yksityiskohdat toisen normaalikartan tueksi. Kun CrazyBump oli generoinut uuden normaalikartan, se yhdistettiin Photoshopissa toiseen karttaan. Valmis normaalikartta tallennettiin .PNG-tiedostomuodossa uuteen dokumenttiin nimellä ”CargoShip_normal”.

9.13 Pelimoottoriin vieminen

Valmis 3D-malli ja tekstuurikartat vietiin lopuksi Unity-pelimoottoriin, jossa niille luotiin uusi materiaali. Assets-valikon alta valittiin Import New Asset -toiminto, jonka

avulla tekstuurikartat tuotiin pelimoottoriin. Kun tekstuurikartat oli tuotu, Assets > Create > Material -toiminnolla luotiin uusi materiaali 3D-mallia varten. Uudelle materiaalille annettiin nimeksi ”CargoShip”.

Materiaali-ikkunassa tekstuurit vietiin omille paikoilleen käyttämällä Select Texture -ikkunan hakutoimintoa. Valmis materiaali raahattiin 3D-mallin päälle kolmiulotteisessa näkymässä, jonka jälkeen assetti oli valmis kenttäsuunnittelijan käytettäväksi. Kuva 54 esittää rahtilaivan 3D-mallia ja materiaalia Unity-pelimoottorin käyttöliittymässä ja kuva 55 3D-mallista renderöityä kuvaa Marmoset Toolbag 2 -ohjelmassa.



KUVA 54. Laivan 3D-malli ja materiaali Unity-pelimoottorissa



KUVA 55. Renderöity kuva laivasta Marmoset Toolbag 2 -ohjelmassa

POHDINTA

Tekstuurigrafiikan laadun kannalta toivottuun tavoitteeseen päästiin kohtalaisesti ja pelin graafinen ilme sai melko hyviä arvosteluita. Grafiikoita kuvailtiin toimiviksi, vaikkakin hieman yksinkertaisiksi, ja pelin sanottiin vastaavan graafisesti sitä tasoa, mitä tämän lajityypin peliltä voitaisiin odottaa.

Ajan säästämiseksi Rescue 2 -projektiin otettiin edellisestä osasta projektin loppua kohden yhä enenevässä määrin vanhoja grafiikka-asetteja, joita sitten päivitettiin vastaamaan uudempaa visuaalista tyyliä. Huonona puolena tässä menettelyssä oli objektien matala polygonimäärä sekä tekstuurien pieni resoluutio. Näitä asetteja yritettiin korjata aikataulun puitteissa sekä geometrian, että tekstuurien osalta, jotta liian huomattavaa eroa uudelleenkäytettyjen ja uusien asettien välille ei syntyisi. Tästä huolimatta vanhojen asettien käyttämät visuaaliset tyylivalinnat kulkeutuivat ainakin osittain myös uuteen peliin asti, joka ei halutun lopputuloksen kannalta ollut erityisen toivottavaa.

Täysin uusien asettien tuottaminen yhtä korkealla laadulla olisi vaatinut huomattavasti enemmän lisätunteja työmäärään, joka käytännössä olisi tarkoittanut lisähenkilökunnan palkkaamista projektiin tai julkaisun viivyttämistä. Tiukasta aikataulusta huolimatta riittävän moni tarvittava asetti saatiin teksturoitua yhtä hyvällä laadulla, kuin muutkin pelin asetit, eikä yleisilmettä rikkovia grafiikoita erottunut liian selkästi peliä pelattaessa.

Grafiikkatiimin pääasiallinen vahvuus oli valtaosan projektista noin kaksi henkeä yhtä aikaisesti. Vain muutama henkilö työskenteli projektissa sen alusta loppuun asti ja valtaosa tiimin jäsenistä vaihtui pitkin projektin elinkaarta. Pysyvä henkilöstö olisi auttanut ajankäytön maksimoimisessa ja tasalaatuisen lopputuloksen aikaansaamisessa, kun graafikkojen aikaavievältä tehtäviin perehdyttämiseltä ja vaihtelevalta tekstuurien laadulta oltaisiin säästyty. Grafiikkatiimin rajallinen koko johti myös siihen, että yksittäisten asettien tekoaikaa jouduttiin vähentämään. Tämä taas johti siihen, että kokeneempien artistien kyvyt tuottaa yksityiskohtaisempia 3D-malleja ja laadukkaampia tekstuurikarttoja eivät päässeet täysin oikeuksiinsa.

Henkilöstön vaihtuvuuden takia uusien artistien työtä helpottamaan olisi tarvittu projektia varten suunniteltu tyyliopas, jossa määritellään hahmo-, ajoneuvo- ja rakennusasettien sisältö. Artistien käytettävissä ollut referenssimateriaali koostui ainoastaan markkinointimateriaaleja varten koostetusta konseptitaiteesta. Lyhyehkön tyylioppaan laatiminen olisi voinut nopeuttaa uusien artistien työskentelyä ja pitää yllä yhtenäistä laatua. Tästä huolimatta grafiikoiden laatu onnistuttiin pitämään melko tasaisena. Tilanteessa, jossa ulkoistamista olisi käytetty, tyylioppaan laatiminen olisi ollut välttämätöntä.

Pelin visuaalista ilmettä oltaisiin voitu parantaa hyödyntämällä erilaisia tekstuurikarttoja sekä uudempaa varjostinteknologiaa. Lukuun ottamatta pelihahmoja, ajoneuvoja sekä muutamia tärkeämpiä asetteja, melko harvassa peliobjektissa käytettiin enempää kuin yhtä tekstuurikarttatyyppeä. Ajanpuutteen vuoksi monen rakennuksen varjostimessa käytettiin vain diffuusiokarttaa. Lisäksi normaalikarttojen tuottaminen jokaiseen asettiin olisi lisännyt syvyyttä ja lisärealismia ympäristöön. Korkean polygonimäärän asettien luominen ja niiden muuttaminen matalapolygoniseen muotoon normaalikarttojen luomiseksi vaatii mallintajalta kuitenkin ylimääräistä aikaa, joten harva malli sai tämän käsittelyn.

PBR-tekniikan käyttäminen pelin varjostimissa olisi ollut myös mahdollinen tapa lisätä pelin realistista ulkoasua ja parantaa sen näyttävyyttä. Valitettavasti projektissa käytetty Unity-pelimoottori esitteli tuen PBR-tekniikan käyttämiselle moottorissa vasta maaliskuussa 2015, versiossa 5.0.0. Tässä vaiheessa peli oli jo lähes julkaisuvaiheessa, joten uutta tekniikkaa ei ehditty ottaa käyttöön. Projektin artisteilla ei myöskään ollut aikaisempaa kokemusta PBR-standardista, joten sen käyttöönotto olisi vaatinut uuden teknologian opettamista koko teksturiartistitiimille. Täysin uuden standardin käyttöönotto olisi hidastanut tuotantoa.

Jos kaikki edellä mainitut ehdotukset olisi toteutettu, projektin valmistuminen aikataulussa olisi ollut mahdotonta. Tästä syystä pelin grafiikoiden tuottaminen olisi ollut järkevämpää ulkoistaa projektin vaatimusten tasolla olevalle yritykselle. Tällöin projektin pääartisti tekisi esimerkkiassetin, jonka jälkeen alihankkijalle lähetettäisiin riittävät viite- ja konseptikuvat halutuista aseteista. Ohjeistuksen perusteella alihankkija toteuttaisi pyydettyt assetit. Tällöin ulkoistus olisi voinut lisätä grafiikoiden tuottamisen tehokkuutta projektissa.

Jatkossa julkaistavien projektien graafista näyttävyyttä olisi siis mahdollista vielä parantaa käyttämällä joko alihankintaa tai kasvattamalla henkilöstöä ja ottamalla käyttöön uudempia tekniikoita. Käytettävissä olevan ajan ja resurssien takia grafiikoiden lopputuloksen kohdalla on tehtävä kuitenkin kompromisseja, jotta pelin ulkoasu saadaan hyväksyttävän näköiseksi ja julkaistua julkaisijan tavoitteiden mukaisesti.

On kuitenkin huomioitava, että lähes jokaisella kaupallisella pelialan julkaisulla on olemassa aina aikataulu sekä budjetti, jotka asettavat haasteita myös pelin graafiselle ulkoasulle. Tästä syystä jokaisessa peliprojektissa on alueita, joiden grafiikkaa oltaisi voitu parantaa.

LÄHTEET

3D Artist. 2016. Quixel Tutorial: Texture Vehicles. Viitattu 22.11.2017.
<https://www.3dartistonline.com/news/2016/06/quixel-tutorial-texture-vehicles/>

3D Modeling 4 Business. 2012. Bump Maps. Viitattu 1.11.2017.
<http://3dmodeling4business.com/bump-map/>

Ahearn, L. 2008. 3D game environments: create professional 3D game worlds. Burlington, MA: Focal Press.

Ahearn, L. 2009. 3D game textures: create professional game art using Photoshop. Boston: Focal Press/Elsevier.

Allegorithmic Documentation. 2017. Thickness Map from Mesh. Viitattu 1.11.2017.
<https://support.allegorithmic.com/documentation/display/SD46/Thickness+Map+from+Mesh>

Allegorithmic. 2017. Substance B2M. Viitattu 1.11.2017.
<https://www.allegorithmic.com/products/bitmap2material>

Allegorithmic. 2017. Substance Painter. Viitattu 1.11.2017.
<https://www.allegorithmic.com/products/substance-painter>

Anhut, A. 2016. Stylized Realism. Viitattu 11.11.2017.
<http://howtonotsuckatgamedesign.com/2016/01/stylized-realism/>

ArtStation. 2015. Torchlight 2, Matthew Tinari. Viitattu 22.11.2017.
<https://www.artstation.com/artwork/wrvkw>

ArtStation. 2016. Dishonored Grounds, Yannick Gombart. Viitattu 22.11.2017.
<https://www.artstation.com/artwork/2YDWK>

Autodesk Mudbox Online Help. 2010. Ambient occlusion maps overview. Viitattu 22.11.2017.

<http://download.autodesk.com/us/mudbox/help2010/index.html?url=WS73099cc142f487551a80ae1b11df600637e-265.htm,topicNumber=d0e17342>

Autodesk. 2013. Learning Autodesk 3ds Max 2008 Foundation. Focal Press.

Birn, J. 2013. Digital Lightning and Rendering, Third Edition. New Riders.

Blender 2.79 Manual. 2017. Ambient Occlusion. Viitattu 1.11.2017.

https://docs.blender.org/manual/de/dev/render/blender_render/world/ambient_occlusion.html

Blender 2.79 Manual. 2017. Displacement Maps. Viitattu 1.11.2017.

https://docs.blender.org/manual/en/dev/render/blender_render/textures/properties/influence/displacement.html

Blender 2.79 Manual. 2017. Procedural Textures. Viitattu 22.11.2017.

https://docs.blender.org/manual/en/dev/render/blender_render/textures/types/procedural/introduction.html

Blender 2.79 Manual. 2017. Structure. Viitattu 1.11.2017.

<https://docs.blender.org/manual/en/dev/modeling/meshes/structure.html>

Blender 2.79 Manual. 2017. UV Overview. Viitattu 1.11.2017.

https://docs.blender.org/manual/en/dev/editors/uv_image/uv/overview.html

Brinck, W. 2009. An In-Depth Look at UVW Mapping an Object. Viitattu 1.11.2017.

<https://www.3dtotal.com/tutorial/1337-an-in-depth-look-at-uvw-mapping-object-3ds-max-by-waylon-brinck?page=2>

Bubenová, Z. 2016. Texturing a 3D Character in Hand-painted Style. Metropolia Metropolia Ammattikorkeakoulu. Opinnäytetyö.

CamingCFG. 2017. Highlands Crossing Diablo 3 screenshot. Viitattu 26.11.2017.
<http://www.gamingcfg.com/screenshot/Highlands-Crossing-Diablo-3-1927>

Chang, C. 2006. Modeling, UV Mapping, and Texturing 3D Game Weapons, Wordware Publishing, Inc.

Chopine, A. 2012. 3D Art Essentials. Burlington: Focal Press.

Coleman, J. 2017. Visual styles In Videogames. Viitattu 1.11.2017.
<https://jahmelcoleman.wordpress.com/games-development/200-2/>

Computer Graphics Library. 2017. MipMap Texturing. Viitattu 1.11.2017.
https://graphics.ethz.ch/teaching/former/vc_master_06/Downloads/Mipmaps_1.pdf

Computer Graphics Wiki. 2017. Texture mapping. Viitattu 1.11.2017.
http://graphics.wikia.com/wiki/Texture_mapping

Creative Bloq Staff. 2013. 10 tips for better Photoshop textures. Viitattu 1.11.2017.
<http://www.creativebloq.com/10-tips-better-photoshop-textures-3133109>

Davis, K. 2013. creating a procedural land texture with images. Viitattu 22.11.2017.
<http://www.proceduraltextures.com/2013/11/21/creating-a-procedural-land-texture-with-images/>

Daz 3D Forums. 2015. Show Us Your Iray Renders. Part II. Viitattu 1.11.2017.
<https://www.daz3d.com/forums/discussion/54734/show-us-your-iray-renders-part-ii/p30>

Demers, O. 2002. Digital Texturing & Painting. New Riders

Department of Radio Engineering. 2017. Selecting a Lighting Method. Viitattu 1.11.2017. <http://radio.feld.cvut.cz/matlab/techdoc/visualize/chlight6.html>

Develop. 2013. Next-gen Snowdrop Engine showcased in new Tom Clancy's The Division trailer. Viitattu 22.11.2017. <http://www.develop-online.net/news/next-gen-snowdrop-engine-showcased-in-new-tom-clancy-s-the-division-trailer/0187043>

Epic Games. 2017. Use the Emissive Material Input. Viitattu 1.11.2017.

<https://docs.unrealengine.com/latest/INT/Engine/Rendering/Materials/HowTo/EmissiveGlow/>

Epic Games. 2017. Using Subsurface Scattering in Your Materials. Viitattu 1.11.2017.

https://docs.unrealengine.com/latest/INT/Engine/Rendering/Materials/HowTo/Subsurface_Scattering/

Epic Games. 2017. Using Texture Masks. Unreal Engine. Viitattu 1.11.2017.

<https://docs.unrealengine.com/latest/INT/Engine/Rendering/Materials/HowTo/Masking/>

Fletcher, J. 2012. A Walkthrough on Stylized Painted Textures. Vertex 1. Luettavissa

<https://gumroad.com/l/GAAOs>

Flickr. 2016. Battlefield 1 7K. Viitattu 22.11.2017.

<https://www.flickr.com/photos/135887904@N06/sets/72157674056752971/with/30329560171/>

Foundry. 2017. Mari Features. Viitattu 1.11.2017.

<https://www.foundry.com/products/mari/features>

Gahan, A. 2012. 3ds Max Modeling for Games: Volume II. Waltham, MA: Focal Press.

Gahan, A. 2013. 3ds Max Modeling for Games, 2nd Edition. Waltham, MA: Focal Press.

GameBanana. 2017. How to edit textures with photoshop. Viitattu 22.11.2017.

<https://gamebanana.com/tuts/12320>

Gantzler, T. 2005. Game Development Essentials: Video Game Art. Cengage Learning.

Goo Create Learn. 2016. Material fundamentals. Viitattu 22.11.2017.

<https://learn.goocreate.com/tutorials/create/material-fundamentals/>

Goodsell, B. 2017. Crazy Bump Tutorial: Understanding Setting for Better Normal Maps. Viitattu 1.11.2017. <https://iamsparky.wordpress.com/crazybump/>

Granberg, C. 2014. David Perry on Game Design: A Brainstorming Toolbox, Cengage Learning.

Hawkins, J. 2013. Here's Why Bioshock Infinite's Stylized Texturing May Be The Key To Its Beauty. Viitattu 1.11.2017. <http://nwn.blogs.com/nwn/2013/04/bioshock-infinities-texturing-may-be-the-key-to-its-beauty.html>

Jaggo, J., Luik, M. & Tunnel, R. 2017. Shading and Lighting. Computer Graphics Learning. Viitattu 1.11.2017. <https://cglearn.codelight.eu/pub/computer-graphics/shading-and-lighting>

Kahrama, J. 2001. Maalaa ja valaise. MikroPC 20/2001, 46. Luettavissa <http://mikropc.net/nettilehti/pdf/pc1312200146.pdf>

Lampel, J. 2017. Normal vs. Displacement Mapping & Why Games Use Normals. Viitattu 1.11.2017. <https://cgcookie.com/articles/normal-vs-displacement-mapping-why-games-use-normals>

MegaSplat. 2017. Tools. Viitattu 22.11.2017. <http://megasplat.com/tools.html>

MotionMedia. 2017. Quixel Suite (academic). Viitattu 1.11.2017. <https://www.motionmedia.com/quixel-suite-academic/>

NVIDIA Developer. 2017. NVIDIA Texture Tools for Adobe Photoshop. Viitattu 1.11.2017. <https://developer.nvidia.com/nvidia-texture-tools-adobe-photoshop>

obedient echo. 2015. knald. Viitattu 22.11.2017. <http://mahribfyp.tumblr.com/post/111684819781/knald>

Orsvärn, L. 2015. Physically based rendering. Wolfire Games Blog. Viitattu 1.11.2017. <http://blog.wolfire.com/2015/10/Physically-based-rendering>

Owen, S. 1999. Shading Models for Scan-Line Graphics. Viitattu 1.11.2017.
<https://www.siggraph.org>

Pandey, J. 2016. Making 3D realistic Tree in AutoCAD with Opacity Map. Viitattu 22.11.2017. <https://thesourcecad.com/making-realistic-tree-in-autocad-with-opacity-map/>

Paquette, A. 2009. Computer Graphics for Artists II: Environments and Characters. Springer Science & Business Media.

PC Magazine Encyclopedia. 2017. Definition of: procedural texture. Viitattu 22.11.2017. <https://www.pcmag.com/encyclopedia/term/49743/procedural-texture>

Perrin, A. 2014. Level of detail. Viitattu 22.11.2017.
<https://ashperrin.wordpress.com/games-development/year-2/unit-71/level-of-detail/>

Petrany, S. 2014. Techniques For Creating Custom Textures In Photoshop. Viitattu 1.11.2017. <https://www.smashingmagazine.com/2014/07/creating-custom-textures-photoshop-techniques/>

Pixar. 2017. Subsurface Scattering. Viitattu 1.11.2017.
<https://renderman.pixar.com/view/DP24192>

Polycount Wiki. 2015. Ambient Occlusion map. Viitattu 1.11.2017.
http://wiki.polycount.com/wiki/Ambient_occlusion_map

Polycount Wiki. 2015. Ambient occlusion vertex color. Viitattu 1.11.2017.
http://wiki.polycount.com/wiki/Ambient_occlusion_vertex_color

Polycount Wiki. 2016. Cube map. Viitattu 1.11.2017.
http://wiki.polycount.com/wiki/Cube_map

Polycount Wiki. 2016. Displacement map. Viitattu 1.11.2017.
http://wiki.polycount.com/wiki/Displacement_map

Polycount Wiki. 2016. Normal map. Viitattu 1.11.2017.
http://wiki.polycount.com/wiki/Normal_map

Polycount. 2010. Quick question about ambient occlusion maps. Viitattu 1.11.2017.
<http://polycount.com/discussion/78649>

Polycount. 2017. Best way to make emissive maps. Viitattu 1.11.2017.
<http://polycount.com/discussion/90648/best-way-to-make-emissive-maps>

Pytte, K. 2016. Research Blog 4 - On the topic of 3D. Viitattu 1.11.2017.
<http://mdu115-kapytte.blogspot.fi/>

Reallusion. 2016. Types of Maps. Viitattu 1.11.2017.
http://www.reallusion.com/iclone/Help/iClone3/15_Multiple_Channel_Texture_Mapping/Types_of_maps.htm

Russel, J. 2015. Basic Theory of Physically-Based Rendering. Marmoset LLC. Viitattu 1.11.2017. <https://www.marmoset.co/posts/basic-theory-of-physically-based-rendering/>

Russell, E. Eliminate Texture Confusion: Bump, Normal and Displacement Maps. Viitattu 1.11.2017. <http://blog.digitaltutors.com/bump-normal-and-displacement-maps/>

Russell, E. Understanding the Difference between Texture Maps. Viitattu 1.11.2017.
<http://blog.digitaltutors.com/understanding-difference-texture-maps/>

Schultz, W. 2017. Creating Realistic Photo Textures for Games - Intro. Viitattu 1.11.2017. <https://www.thoughtco.com/creating-realistic-photo-textures-for-games-139397>

SG Gaming Info. 2013. Black Ops 2 makes an explosive launch trailer. Viitattu 22.11.2017. <http://www.sggaminginfo.com/2012/10/black-ops-2-makes-an-explosive-launch-trailer/>

Sketchfab Blog. 2015. Art Spotlight: Main Frame - Hacking Terminal. Viitattu 22.11.2017. <https://blog.sketchfab.com/art-spotlight-main-frame-hacking-terminal/>

Slick, J. 2016. 3D Model Components - Vertices, Edges, Polygons & More. Viitattu 1.11.2017. <https://www.lifewire.com/3d-model-components-1952>

Slick, J. 2016. Surfacing 101: Creating a UV Layout. Viitattu 1.11.2017. <https://www.lifewire.com/creating-a-uv-layout-1955>

Smith, G. 2013. 3D Workflow for Designers & Art Directors: Modeling. Viitattu 1.11.2017. <http://blog.hitchingsdesign.com/3d-workflow-for-designers-art-directors-modeling/>

Splash Damage Wiki. 2007. Specular Maps. Viitattu 1.11.2017. http://wiki.splashdamage.com/index.php/Specular_Maps

Steam. 2017. RESCUE 2: Everyday Heroes. Viitattu 22.11.2017. http://store.steampowered.com/app/339210/RESCUE_2_Everyday_Heroes/

Steam. 2017. SUPERHOT. Viitattu 22.11.2017. <http://store.steampowered.com/app/322500/SUPERHOT/>

The Gnomon Workshop. 2017. Introduction to Maxon BodyPaint 3D. Viitattu 22.11.2017. <https://www.thegnomonworkshop.com/tutorials/introduction-to-maxon-bodypaint-3d>

Tovar, I. 2016. 3D scanning tech drives major success in Battlefield 1. Viitattu 22.11.2017. <https://3dprintingindustry.com/news/3d-scanning-tech-drives-major-success-battlefield-1-97353/>

UE4 AnswerHub. 2015. Question about PBR textures. Viitattu 22.11.2017. <https://answers.unrealengine.com/questions/295943/question-about-pbr-textures.html>

Unity Technologies. 2014. Unity - Manual: Materials and Shaders. Viitattu 1.11.2017. <https://docs.unity3d.com/455/Documentation/Manual/Materials.html>

Unity Technologies. 2017. A Gentle Introduction to Shaders. Viitattu 22.11.2017.
<https://unity3d.com/learn/tutorials/topics/graphics/gentle-introduction-shaders>

Unity Technologies. 2017. Creating and Using Materials. Viitattu 1.11.2017.
<https://docs.unity3d.com/Manual/Materials.html>

Unity Technologies. 2017. Unity - Manual: Cubemap. Unity-pelimoottorin Internet-manuaali. Viitattu 30.5.2017. <https://docs.unity3d.com/Manual/class-Cubemap.html>

Unity Technologies. 2017. Unity - Manual: Materials, Shaders & Textures. Viitattu 1.11.2017. <https://docs.unity3d.com/Manual/Shader.html>

VALVE Developer Community, 2015. MIP Mapping. Viitattu 1.11.2017.
https://developer.valvesoftware.com/wiki/MIP_Mapping

Wilson, J. 2014. Complex Material Setup. Marmoset LLC. Viitattu 1.11.2017.
<https://www.marmoset.co/posts/complex-material-setup/>

Wilson, J. 2015. PBR Texture Conversion. Marmoset LLC. Viitattu 1.11.2017.
<https://www.marmoset.co/posts/pbr-texture-conversion/>

Wilson, J. 2015. Physically-Based Rendering, And You Can Too!. Marmoset LLC. Viitattu 1.11.2017. <https://www.marmoset.co/posts/physically-based-rendering-and-you-can-too/>